PB80-191109

Users Manual for Hydrological Simulation Program FORTRAN (HSPF)

Hydrocomp, Inc.
Mountain View, CA

Prepared for

Environmental Research Lab.
Athens, GA

Apr 80

U.S. Department of Commerce
National Technical Information Service
NTIS

United States
Environmental Protection
Agency

Environmental Research
Laboratory
Athens GA 30605

EPA-600/9-80-015
April 1980

PB80-191109

Research and Development

EPA

Users Manual for Hydrological Simulation Program-Fortran (HSPF)

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U. S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

RESEARCH REPORTING SERIES

Research reports of the Office of Research and Development, U.S. Environmental Protection Agency, have been grouped into nine series. These nine broad categories were established to facilitate further development and application of environmental technology. Elimination of traditional grouping was consciously planned to foster technology transfer and a maximum interface in related fields. The nine series are:

1. Environmental Health Effects Research
2. Environmental Protection Technology
3. Ecological Research
4. Environmental Monitoring
5. Socioeconomic Environmental Studies
6. Scientific and Technical Assessment Reports (STAR)
7. Interagency Energy-Environment Research and Development
8. "Special" Reports
9. Miscellaneous Reports

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-600/9-80-015	2.	3. RECIPIENT'S ACCESSION NO. PB80 191109
4. TITLE AND SUBTITLE Users Manual for Hydrological Simulation Program - FORTRAN (HSPF)		5. REPORT DATE April 1980 issuing date
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Robert C. Johanson, John C. Imhoff, and Harley H. Davis, Jr.		8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Hydrocomp, Incorporated 201 San Antonio Circle Mountain View, California 94040		10. PROGRAM ELEMENT NO. A28B1A
		11. CONTRACT/GRANT NO. R804971-01
12. SPONSORING AGENCY NAME AND ADDRESS Environmental Research Laboratory--Athens GA Office of Research and Development U.S. Environmental Protection Agency Athens, Georgia 30605		13. TYPE OF REPORT AND PERIOD COVERED Final, 11/76-11/78
		14. SPONSORING AGENCY CODE EPA/600/01
15. SUPPLEMENTARY NOTES		

16. ABSTRACT

The Hydrological Simulation Program - FORTRAN (HSPF) is a set of computer codes that can simulate the hydrologic,and associated water quality, processes on pervious and impervious land surfaces and in streams and well-mixed impoundments. The manual discusses the modular structure of the system, the principles of structured programming technology, and the use of these principles in the construction of the HSPF software. In addition to a pictorial representation of how each of the 500 subprograms fits into the system, the manual presents a detailed discussion of the algorithms used to simulate various water quality and quantity processes. Data useful to those who need to install, maintain, or alter the system or who wish to examine its structure in greater detail are also presented.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Simulation		12A
Water quality		68D

18. DISTRIBUTION STATEMENT RELEASE TO PUBLIC	19. SECURITY CLASS *(This Report)* UNCLASSIFIED	21. NO. OF PAGES 684
	20. SECURITY CLASS *(This page)* UNCLASSIFIED	22. PRICE

EPA Form 2220-1 (9-73)

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980-657-146/5644

EPA-600/9-80-015
April 1980

USERS MANUAL FOR HYDROLOGICAL SIMULATION PROGRAM - FORTRAN (HSPF)

by

Robert C. Johanson
John C. Imhoff
Harley H. Davis, Jr.
Hydrocomp Incorporated
Mountain View, California 94040

Grant No. R804971-01

Project Officer

Thomas O. Barnwell
Technology Development and
Applications Branch
Environmental Research Laboratory
Athens, Georgia 30605

ENVIRONMENTAL RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
ATHENS, GEORGIA 30605

DISCLAIMER

This report has been reviewed by the Environmental Research Laboratory, U.S. Environmental Protection Agency, Athens, Georgia and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the U.S. Environmental Protection Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

FOREWORD

As environmental controls become more costly to implement and the penalties of judgment errors become more severe, environmental quality management requires more efficient analytical tools based on greater knowledge of the environmental phenomena to be managed. As part of this Laboratory's research on the occurrence, movement, transformation, impact, and control of environmental contaminants, the Technology Development and Applications Branch develops management or engineering tools to help pollution control officials achieve water quality goals through watershed management.

The development and application of mathematical models to simulate the movement of pollutants through a watershed and thus to anticipate environmental problems has been the subject of intensive EPA research for several years. The most recent advance in this modeling approach is the Hydrological Simulation Program - FORTRAN (HSPF), which uses digital computers to simulate hydrology and water quality in natural and man-made water systems. HSPF is designed for easy application to most watersheds using existing meteorologic ahd hydrologic data. Although data requirements are extensive and running costs are significant, HSPF is thought to be the most accurate and appropriate management tool presently available for the continuous simulation of hydrology and water quality in watersheds.

David W. Duttweiler
Director
Environmental Research Laboratory
Athens, Georgia

ABSTRACT

The Hydrological Simulation Program - FORTRAN (HSPF) is a set of computer codes that can simulate the hydrologic, and associated water quality, processes on pervious and impervious land surfaces and in streams and well-mixed impoundments. The manual discusses the modular structure of the system, the principles of structured programming technology, and the use of these principles in the construction of the HSPF software. In addition to a pictorial representation of how each of the 500 subprograms fits into the system, the manual presents a detailed discussion of the algorithms used to simulate various water quantity and quality processes. Data useful to those who need to install, maintain, or alter the system or who wish to examine its structure in greater detail are also presented.

The report was submitted in fulfillment of Grant No. R804971-01 by Hydrocomp, Inc., under the sponsorship of the U.S. Environmental Protection Agency. The report covers the period November 1, 1976, to November 30, 1978, and work was completed as of January 16, 1980.

CONTENTS

Foreword iii
Abstract iv

Part

A Introduction 1
B General Principles 8
C Standards and Conventions 23
D Visual Table of Contents 45
E Functional Description 130
F Format for the Users Control Input 320

Appendices

I Glossary of Terms 635
II Sample Runs 642
III Program NEWTSS 656
IV Guide to the Programmers Supplement 674

PART A INTRODUCTION

CONTENTS

1.0 Purpose and Scope of the HSPF Software 2
2.0 Requirements for HSPF 4
3.0 Purpose and Organization of this Document 5
4.0 Definition of Terms 6
5.0 Notice of User Responsibility 6
6.0 Acknowledgments 6

1.0 PURPOSE AND SCOPE OF THE HSPF SOFTWARE

The use of models which simulate continuously the quantity/quality processes occurring in the hydrological cycle is increasing rapidly. Recently there has been a proliferation in the variety of models and in the range of processes they simulate. This has been a mixed blessing to a user. To get the benefits of simulation, he has to select a model from a bewildering array and then spend much effort amassing and manipulating the huge quantities of data which the model requires. If he wishes to couple two or more subprocess models to simulate a complete process, he often encounters further difficulties. The underlying assumptions and/or structures of the subprocess models may make them somewhat incompatible. More frequently, the data structures are so different that coupling requires extensive data conversion work.

One reason for these problems is that the boom in modeling work has not included enough work on the development of good model structures. That is, very few software packages for water resource modeling are built on a systematic framework in which a variety of process modules can fit.

With HSPF we have attempted to overcome these problems as far as possible. HSPF consists of a set of modules arranged in a hierarchical structure, which permit the continuous simulation of a comprehensive range of hydrologic and water quality processes. Our experience with sophisticated models indicates that much of the human effort is associated with data management. This fact, often overlooked by model builders, means that a successful comprehensive model must include a sound data management component. Otherwise, the user may become so entangled in data manipulation that his progress on the simulation work itself is drastically retarded. Consequently, the HSPF software is planned around a time series management system operating on direct access principles. The simulation modules draw input from a Time Series Store and are capable of writing output to it. Because these transfers require very few instructions from the user, the problems referred to above are minimized.

The system is designed so that the various simulation and utility modules can be invoked conveniently, either individually or in tandem. A top down approach emphazing structured design has been followed. First, the overall framework and the Time Series Management System were designed. Then, work progressed down the structure from the highest, most general level to the lowest, most detailed one. Every level was planned before the code was written. Uniform data structures, logic figures, and programming conventions were used throughout. Modules were separated according to function so that, as much as possible, they contained only those activities which are unique to them. Structured design has made the system relatively easy to extend, so that users can add their own modules with relatively little disruption of the existing code.

Now, a note on the initial contents of the system. Presently, it includes modules which can handle almost all the functions which are available in the following existing models:

(1) HSP (LIBRARY, UTILITY, LANDS, CHANNEL, QUALITY)
(2) ARM
(3) NPS

The HSPF software is not merely a translation of the above models, but a new system with a framework designed to accomodate a variety of simulation modules; the modules described above are the initial contents. Many extensions have been made to the above models in the course of restructuring them into the HSPF system.

It is hoped that HSPF will become a valuable tool for water resource planners. Because it is more comprehensive than most existing systems, it should permit more effective planning. More specifically, the package can benefit the user in the following ways:

(1) The time-series-oriented direct access data system and its associated modules can serve as a convenient means of inputting, organizing, and updating the large files needed for continuous simulation.

(2) The unified user oriented structure of the model makes it relatively simple to operate. The user can select those modules and options that he wishes to execute in one run, and the system will ensure that the correct sets of code are invoked and that internal and external transfers of data are handled. This is achieved with a minimum of manual intervention. Input of control information is simplified because a consistent system is used for this data for all the modules.

(3) Because the system has been carefully planned using modern top-down programming techniques, it is relatively easy to modify and extend. The use of uniform programming standards and conventions has assisted in this respect.

(4) Since the code is written almost entirely in ANSI standard Fortran, implementation on a wide variety of machines is possible.

2.0 REQUIREMENTS FOR HSPF

In awarding the grant for development of HSPF, the EPA set the following requirements:

(1) It must manage and perform deterministic simulation of a variety of aquatic processes which occur on and under land surfaces and in channels and reservoirs.

(2) It must readily accommodate alternate or additional simulation modules.

(3) It must permit easy operation of several modules in series, and thus be capable of feeding output from any operation to subsequent operations.

(4) It must be in ANSI Fortran with minor specified extensions.

With the concurrence of the EPA, we expanded on these requirements:

(1) It must have a totally new design. Existing modules should not merely be translated, but should be fitted into a new framework.

(2) It must be designed from the top down, using some of the new improved programming techniques, such as Structured Design and Structured Programming.

(3) Duplication of blocks of code which perform similar or identical functions should be avoided.

(4) The user's control input must have a logically consistent structure throughout the package.

(5) Uniform standards and practices must be followed throughout the design, development and documentation of the system.

(6) It must have a conveniently operated disk-based Time Series Store built on the principle of direct access.

(7) The design must be geared to implementation on larger models of the current generation of "minicomputers." It must be compatible with Operating Systems which share machine core space using either the virtual memory approach or a conventional overlay technique.

3.0 PURPOSE AND ORGANIZATION OF THIS DOCUMENT

This report contains all the documentation of the HSPF system. It is designed to:

(1) introduce new users to the principles and concepts on which the system is founded

(2) describe the technical foundations of the algorithms in the various application (simulation) modules

(3) describe the input which the user supplies to run the system

To meet these needs and, at the same time, to produce a document which is reasonably easy to use, we have divided this report into several distinct parts, each with its own organization and table of contents.

Part A (this one) contains introductory material.

Part B outlines the general principles on which the HSPF system is based. This includes a discussion of the "world view" which our simulation modules embody. A firm grasp of this material is necessary before the detailed material can be properly understood.

Part C explains the standards and conventions which we employed to develop the HSPF software. It discusses the advantages of Structured Programming and describes how we implemented this technique in standard Fortran.

Part D is a visual table of contents. It displays pictorially the entire hierarchy of subprograms, starting with the MAIN program and proceeding down the "tree" to the most detailed level of the system. It gives the name, number and a brief description of each of the 500 subprograms.

Part E documents the function of each part of the software. The organization of this part follows the layout of the software itself. The relationship between, and the functions of, the various modules are described, starting at the highest most general level and proceeding down to the lowest most detailed level following the numbering system in the visual table of contents. The algorithms used to simulate the quantity and quality processes which occur in the real world are described in this part.

Part F describes the User's Control Input; that is, the information which the user must provide in order to run HSPF.

Material which might obscure the structure of this document if it were included in the body of the report appears in Appendices. These include a glossary of terms and sample runs.

4.0 DEFINITION OF TERMS

In this document, terms which have a special meaning in HSPF, are enclosed in quotes the first time they occur. Usually an explanation follows immediately. A glossary of terms will be found in Appendix I.

5.0 NOTICE OF USER RESPONSIBILITY

This product has been carefully developed. Although the work included testing of the software, the ultimate responsibility for its use and for ensuring correctness of the results obtained, rests with the user.

The EPA and the developers of this software make no warranty of any kind with regard to this software and associated documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. They shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

While we intend to correct any errors which users report, we are not obliged to do so. We reserve the right to make a reasonable charge for work which is performed for a specific user at his request.

6.0 ACKNOWLEDGMENTS

This work was sponsored by the Environmental Reasearch Laboratory in Athens, Georgia. David Duttweiler is the laboratory director and Robert Swank the head of the Technology Development and Applications Branch, which supervised the project. The Project Officers were Jim Falco and, later, Tom Barnwell.

Many people at Hydrocomp Inc. were involved in the project . The role of each of the of the major contributors is summarized below. Persons are listed approximately in order of time spent on the project, but this does not necessarily reflect the relative value of their contributions.

Robert Johanson was Project Manager. He was responsible for project coordination, development of the standards and practices and much of the basic design work. He supervised closely the development of the application modules and wrote the SNOW and PWATER sections of the PERLND module, and the HYDR section of the RCHRES module. He was also responsible for the Run Interpreter.

John Imhoff worked on the RCHRES module. He analyzed the HSP QUALITY code, performed the detailed design of the new module and wrote the code and documentation for it. He also coordinated the production of the functional descriptions (Part E) of all the application modules.

Harley Davis designed and coded most sections of the PERLND module and all of the IMPLND module. He also wrote the functional descriptions of those modules.

Janice Walters worked on the Time Series Management System (TSMS), particularly the TSGET and TSSMGR groups of subprograms.

Gerard Lum had the tedious, but critical, task of translating most of the pseudo code into ANSI standard Fortran.

Delbert Franz participated in the overall design of the system. He also supervised the work on the TSMS, and produced most of the code for the TSPUT group of subprograms.

Tamas Eger did the detailed design and coding of most of the TSSMGR subroutine group. He also wrote the stand-alone program NEWTSS.

Carolyn Karnos had the onerous task of assembling the Fortran subprograms into the program file and contending with the "bugs" which arose in this process.

Jean-Jaques Heler participated in the design of the system, especially in shaping the concepts discussed in Part B. He also finished and tested the subprogram groups TSSMGR and NEWTSS after Tamas Eger had to return to Hungary.

Norman Crawford participated in the initial design of the system. As a result, many of his ideas for the HSP-II system found their way into HSPF.

Eleanor Bassler worked on the design and organization of the Users Control Input.

Alan Schiffer translated many of the RCHRES subprograms from pseudo to Fortran code.

Diana Allred edited much of the text in this document, especially Part E. Donna Mitchell also contributed to the editing effort.

Tony Donigian reviewed the text in Part E for technical accuracy.

Nancy Sharpe spent many hours at the copying machine, producing the working documentation from which this document evolved.

Jack Kittle assisted in assembling the code into the program file, and also set up the system for arranging data in the numerous versions of the COMMON block.

Kevin Gartner translated some pseudo code into Fortran.

PART B GENERAL PRINCIPLES

CONTENTS

1.0 View of the Real World 9
- 1.1 General Concepts 9
- 1.2 Nodes, Zones, and Elements 9
- 1.3 Processing Units and Networks 11

2.0 Software Structure 14
- 2.1 Concept of an "Operation" 14
- 2.2 Time Series Storage 16
- 2.3 Times Series Management for an Operation 17
- 2.4 HSPF Software Hierarchy 17

3.0 Structure of a Job 20
- 3.1 Elements of a Job 20
- 3.2 Groups of Operations 20

FIGURES

Number Page

1-1 Nodes, zones and elements 10
1-2 Directed and non-directed graphs 12
1-3 Single- and multi-element processing units 13
2-1 Logical structure of the internal scratch pad 15
2-2 Activities involved in an operation 18
2-3 Overview of HSPF software 19
3-1 Schematic of data flow and storage in a single run 21
3-2 Extract from typical User's Control Input, showing how grouping of operations is specified 22

1.0 VIEW OF THE REAL WORLD

1.1 General Concepts

To design a comprehensive simulation system, one must have a consistent means of representing the prototype; in our case, the real world. We view it as a set of constituents which move through a fixed environment and interact with each other. Water is one constituent; others are sediment, chemicals, etc. The motions and interactions are called processes.

1.2 Nodes, Zones, and Elements

The prototype is a continuum of constituents and processes. Simulation of such a system on a digital computer requires representation in a discrete fashion. In general, we do this by subdividing the prototype into "elements" which consist of "nodes" and "zones."

A node corresponds to a point in space. Therefore, a particular value of a spatially variable function can be associated with it, for example, channel flow rate and/or flow cross sectional area. A zone corresponds to a finite portion of the real world. It is usually associated with the integral of a spatially variable quantity, for example, storage in a channel reach. The zone the smallest unit into which we subdivide the world. The relationship between zonal and nodal values is similar to that between the definite integral of a function and its values at the limits of integration.

An element is a collection of nodes and/or zones. Figure 1-1 illustrates these concepts. We simulate the response of the land phase of the hydrological cycle using elements called "segments." A segment is a portion of the land assumed to have areally uniform properties. A segment of land with a pervious surface is called a "Pervious Land-segment" (PLS). Constituents in a PLS are represented as resident in a set of zones (Fig. 1-1a). A PLS has no nodes. As a further example, consider our formulation of channel routing. We model a channel reach as a one dimensional element consisting of a single zone situated between two nodes (Fig. 1-1b). We simulate the flow rate and depth at the nodes; the zone is associated with storage.

The conventions of the finite element technique also fall within the scope of these concepts. Figure 1-1c shows a two dimensional finite element used in the simulation of an estuary. Three nodes define the boundaries of the triangular element. A fourth node, situated inside, may be viewed as subdividing the element into three zones. This last type of element is not presently used in any HSPF module, but is included in this discussion to show the generality provided by HSPF. The system can accommodate a wide variety of simulation modules.

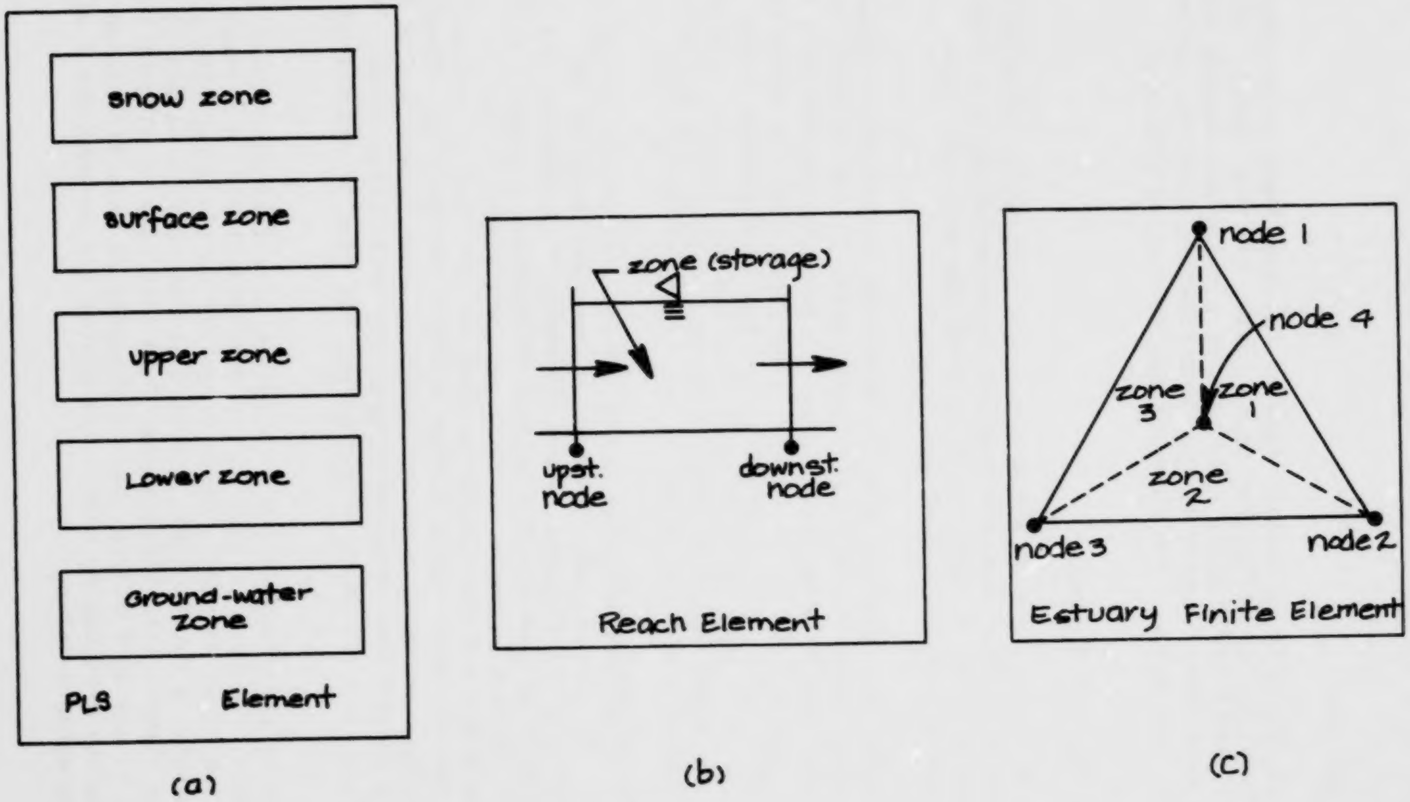


Fig 1-1 Nodes, zones and elements

There are no fixed rules governing the grouping of zones and nodes to form elements. The model builder must decide what grouping is reasonable and meaningful, based on his view of the real world processes being simulated. In the foregoing material we presented some elements used in HSP and other systems. In general, it is convenient to define elements so that a large portion of the real world can be represented by a collection of conceptually identical elements. In this way, a single parameter structure can be defined which applies to every element in the group. Thus, each element is a variation on the basic theme. It is then meaningful to speak of an "element type." For example, elements of type "PLS" all embody the same arrangement of nodes and are represented by sets of parameters with identical structure. Variations between segments are represented only by variations in the values of parameters. The same applies to any other element, such as a Reach, layered lake or a triangular finite element.

As illustrated in the above discussion, nodes are often used to define the boundaries of zones and elements. A zone, characterized by storage, receives inflows and disperses outflows; these are called "fluxes." Note that if the nodal values of a field variable are known, it is often possible to compute the zonal values (storages). The reverse process does not work.

1.3 Processing Units and Networks

To simulate a prototype we must handle the processes occurring within the elements and the transfer of information and constituents between them. The simulation of large prototypes is made convenient by designing a single "application module" for a given type of element or element group, and applying it repetitively to all similar members in the system. For example, we may use the RCHRES module to simulate all the reaches in a watershed using storage routing. This approach is most efficient computationally if one element or group of elements, called a "processing unit" (PU), is simulated for an extended period of time before switching to the next one. To permit this, we must be able to define a processing sequence such that all information required by any PU comes from sources external to the system or from PU's already simulated. This can only happen if the PU's and their connecting fluxes form one or more networks which are "directed graphs." In a directed graph there are no bidirectional paths and no cycles. Figure 1-2 shows some directed and nondirected graphs.

The requirement that PU's form directed graphs provides the rule for grouping elements into PU's. Any elements interacting with each other via loops or bidirectional fluxes must be grouped into a single PU because none of them can be simulated apart from the others.

Thus, we can have both single element and multielement PU's. A PLS is an example of the former and a channel network simulated using the full equations of flow exemplifies the latter (Fig. 1-3). A multielement PU is also known as a "feedback region." The collection of PU's which are simulated in a given run is called a "network."

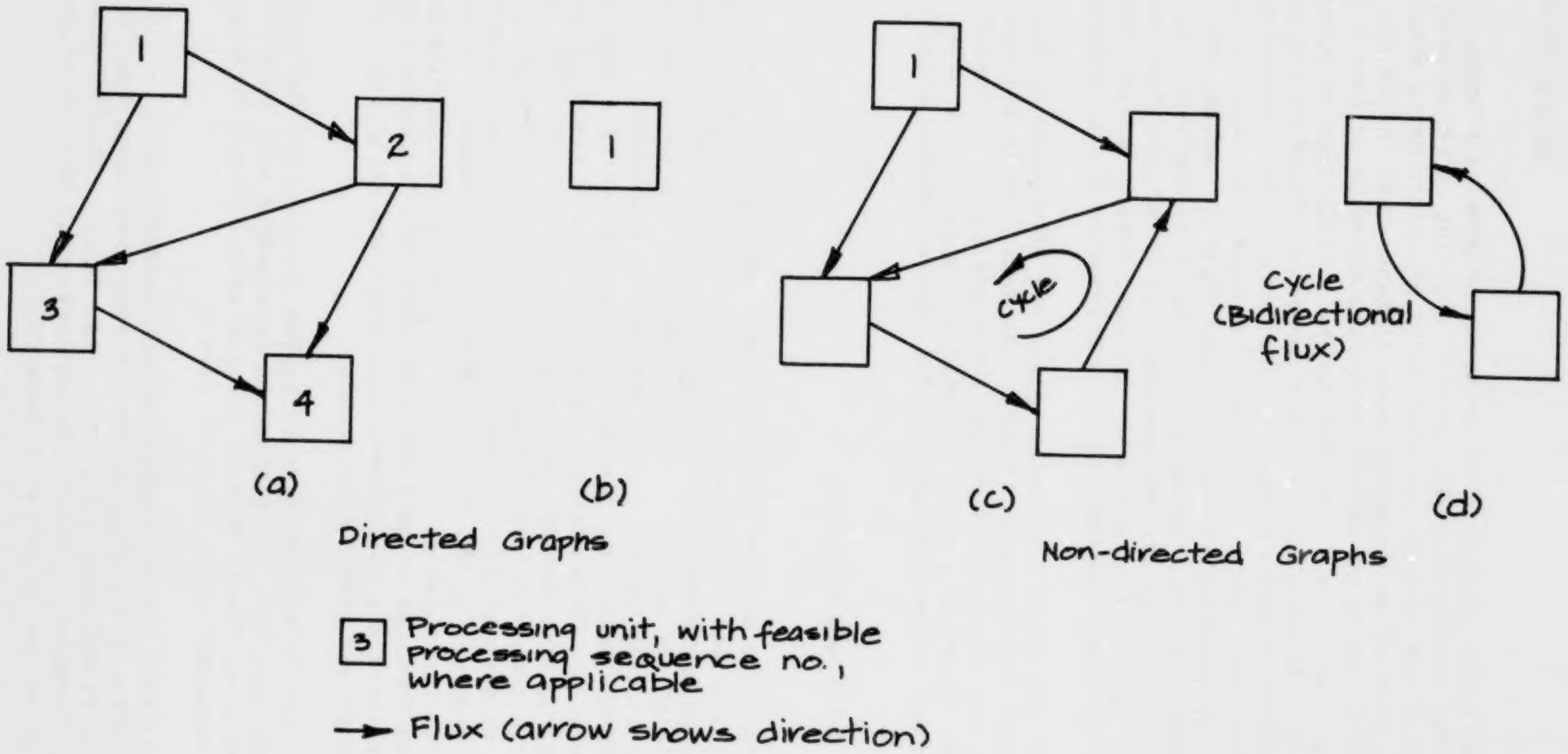

1
2
3
4
1
1
cycle
Cycle
(Bidirectional
flux)
(a)
(b)
(c)
(d)
Directed Graphs
Non-directed Graphs
3 Processing unit, with feasible processing sequence no., where applicable
Flux (arrow shows direction)

Fig. 1-2 Directed and Non-directed Graphs

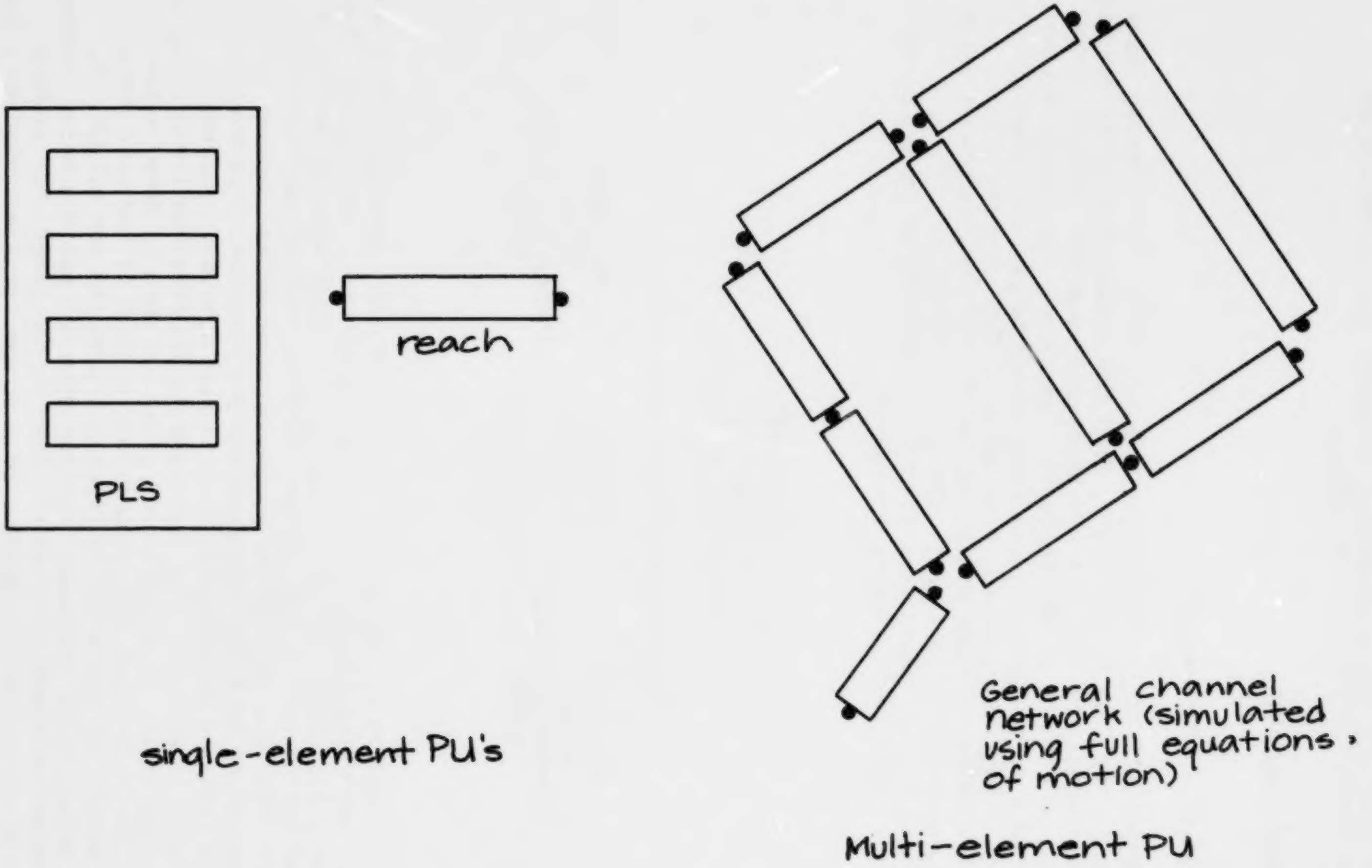

PLS
reach
single-element PU's
General channel network (simulated using full equations, of motion)
Multi-element PU

FIG. 1-3 Single-and multi-element processing units

The processes which occur within a PU are represented mathematically in an "application model." The corresponding computer code is called an "application module" or "simulation module."

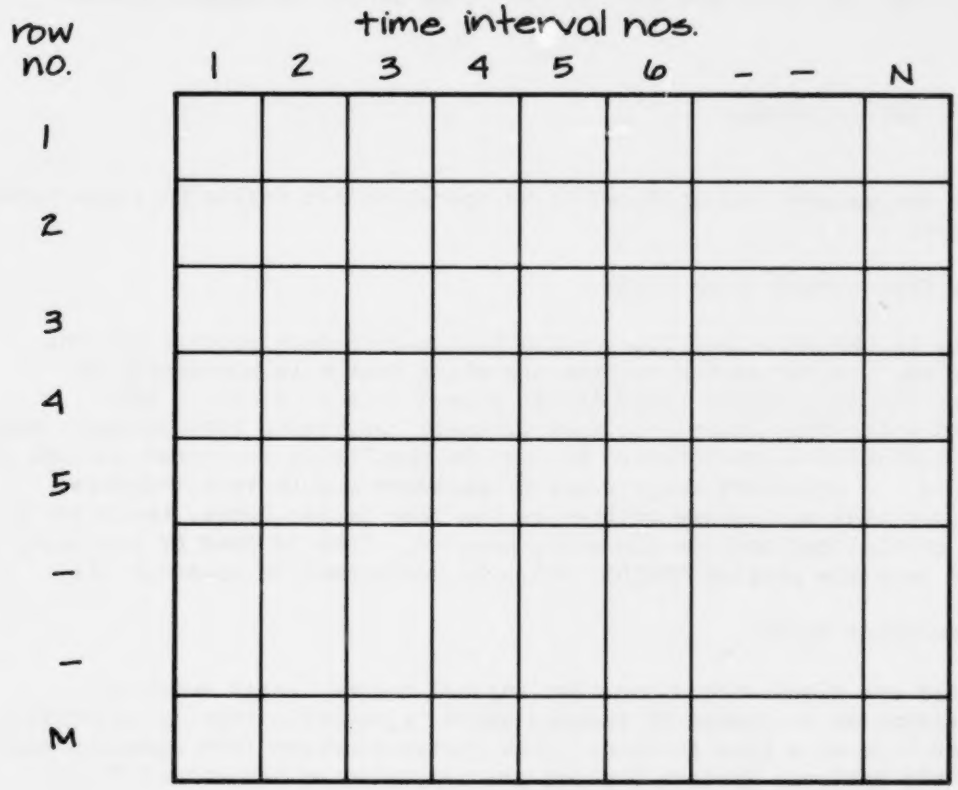
2.0 SOFTWARE STRUCTURE

2.1 Concept of an "Operation"

A great variety of activities are performed by HSPF; for example, input a time series to the Time Series Store, find the cross correlation coefficient for two time series, or simulate the processes in a land segment. They all incorporate two or more of the following functions: get a set of time series, operate on the set of input time series to produce other time series, and output the resulting time series. This applies both to application modules (already discussed) and to "utility modules," which perform operations ancillary or incidental to simulation. Thus, a simulation run may be viewed as a set of "operations" performed in sequence. All operations have the following structure:

```
                          SUPERVISE
                          OPERATIONS
                      (subroutine OSUPER)
                               ¦
             -----------------------------------
             ¦                 ¦                ¦
         GET TIME           OPERATE          PUT TIME
          SERIES            (utility          SERIES
    (subroutine group          or        (subroutine group
          TSGET)          application          TSPUT)
                             module)
```

The OPERATE function is the central activity in the operation. This work is done by an "operating module" (OM) and its subordinate subprograms. They operate for a specified time on a given set of input time series and produce a specified set of output time series, under control of the "operations supervisor" (OSUPER). All of the pieces of time series involved in this internal operation have the same interval and duration. They are therefore viewed as written on an "internal scratch pad" (INPAD), resident in the core of the machine (Fig. 2-1). The operating module receives the scratch pad with some rows filled with input and, after its work is done, returns control to the supervisor with another set of rows filled with output. The operating module may overwrite an input row with its own output. The computing module being executed, together with the options being invoked, will determine the number of rows required in the INPAD. For example, simulation of the hydraulic behavior of a stream requires relatively few time series (eg. inflow, depth and outflow) but the inclusion of water quality simulation adds many more time series to the list. Now, the total quantity of machine space available for storage of time series is also fixed (specified in a COMMON block) by the options in effect; this is the size ("area") of the INPAD. Since both the size (N*M) and number of rows (M) in the INPAD are known, the "width" (no. of intervals,N) can be found. The corresponding physical time is called the "internal scratch pad span (INSPAN)."



NOTE: there is one time series per row.

FIG. 2-1 Logical structure of the internal scratch pad

The "get time series" function prepares the input time series. This work is done by a subroutine group called TSGET. It obtains the correct piece of a time series from the appropriate file, aggregates or disaggregates it to the correct time interval, multiplies the values by a user specified constant (if required), and places the data in the required row of the internal scratch pad. Subroutine group TSPUT performs the reverse set of operations. TSGET and TSPUT are sometimes bypassed if a required time series is already in the INPAD when the operation is started, or if the output is being passed to the next operation via the internal scratch pad.

Modules TSGET and TSPUT are part of the "time series management system" (TSMS).

2.2 Time Series Storage

The time series used and produced by an operation can reside in three types of storage.

(1) The Time Series Store (TSS)

This is the principal library for medium-long term storage of time series. As far as the machine operating system is concerned, it consists of a single large direct access file on a disc. HSPF subdivides this space into many datasets containing time series. Each is logically self-contained but may be physically scattered through the store. A directory keeps track of datasets and their attributes. Before time series are written to the Time Series Store, the store must be initialized and its directory created. This is done by executing the separate program NEWTSS, which is documented in Appendix III.

(2) Sequential Files

These are files with a constant logical record length which are resident on or routed to punched cards, a magnetic tape, a sequential disk file or a line printer. Time series received from agencies such as the National Weather Service are typically on sequential files (cards or tape).

(3) Internal Scratch Pad (INPAD)

If two or more operations performed in sequence use the same internal time step, time series may be passed between them via the INPAD. Successive operations may simply pick up the data written by the previous ones, without any external (disc) transfer taking place. This is typically done when time series representing the flow of water (and constituents) are routed from one stream reach to the one next downstream.

2.3 Time Series Management For An Operation

Any operation involves a subset of the activities shown in Fig. 2-2. The operating module expects a certain set of time series in the INPAD. The operations supervisor, acting under user control, ensures that the appropriate input time series are loaded from whichever source has been selected, and informs the computing module of the rows in the INPAD where it will find its input. Similar arrangements hold for output of time series.

2.4 HSPF Software Hierarchy

The hierarchy of functions in HSPF is shown in Fig. 2-3. Some explanatory notes follow.

The "Run Interpreter" is the group of subprograms which reads and interprets the "Users Control Input." It sets up internal information instructing the system regarding the sequence of operations to be performed. It stores the initial conditions and the parameters for each operation in the appropriate file on disc and creates an instruction file which will ensure that time series are correctly passed between operations, where necessary.

The "TSS management" modules are those used to create, modify, or remove data sets in the time series store.

The "Operations Supervisor" is a subroutine which acts on information provided by the Run Interpreter, invoking the appropriate "application" or "utility" modules. It provides them with the correct values for parameters and state variables by reading the files created by the Run Interpreter.

Operating modules are either "application modules" or "utility modules." They perform the operations which make up a run. Each time one of those modules is called, an operation is performed for a period corresponding to the span of the internal scratch pad (INSPAN). The Operations Supervisor ensures that the correct module is invoked.

"Service subprograms" perform tasks such as reading from and writing to time series storage areas, adding T minutes to a given date and time, to get a new date and time, etc.

The "Time Series Management System" (TSMS) consists of all the modules which are only concerned with manipulation of time series or the files used to store time series. It includes the TSS management functions, and TSGET and TSPUT.

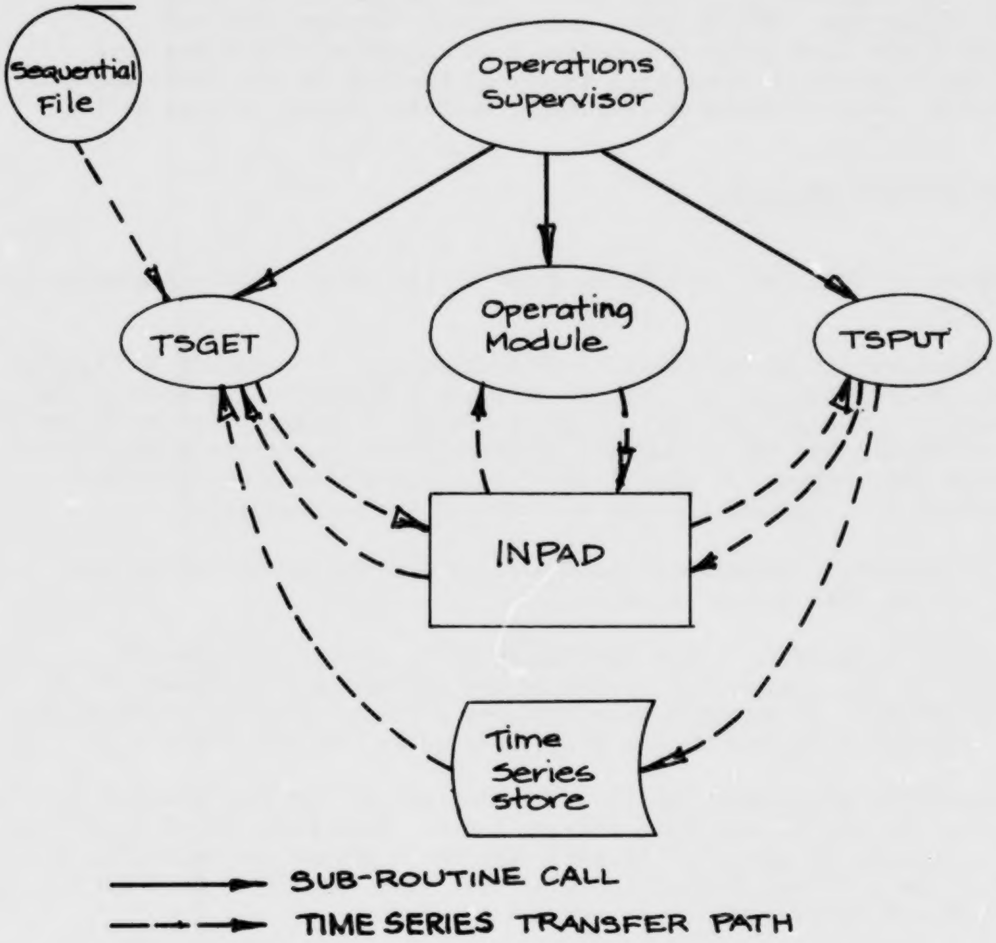
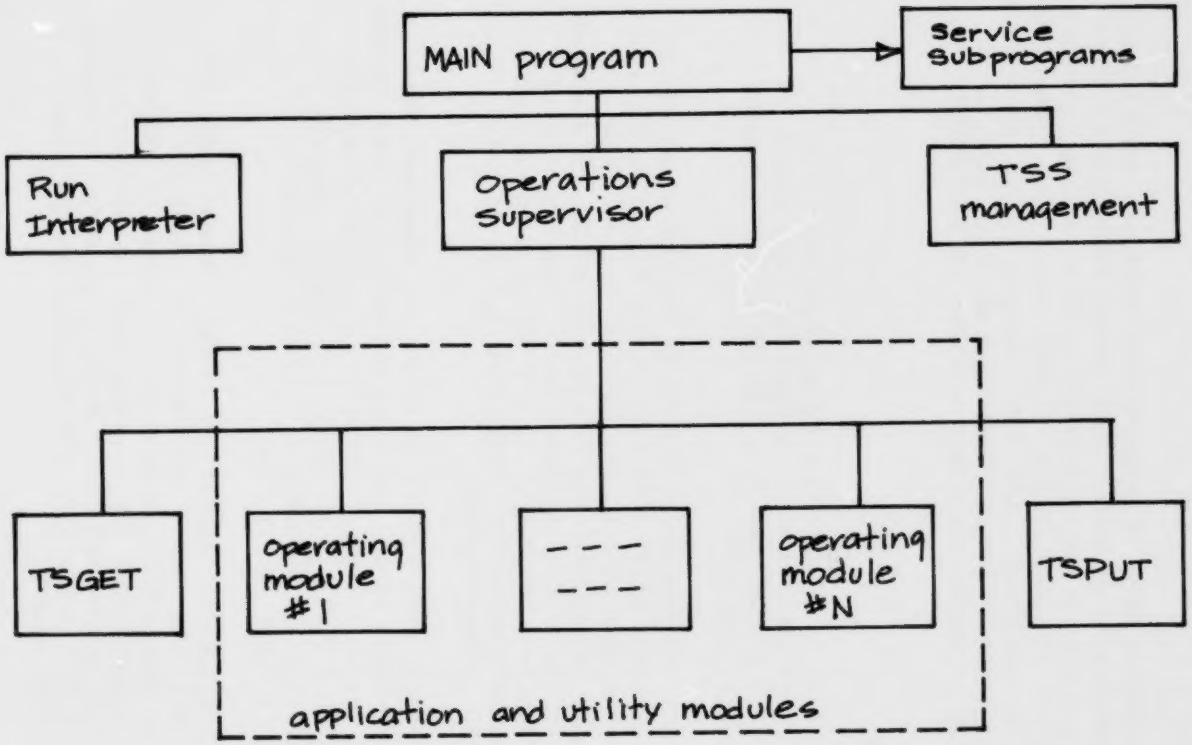


FIG.2-2 ACTIVITIES INVOLVED IN AN OPERATION



TSS Time Series Store

Fig. 2-3 Overview of HSPF software

3.0 STRUCTURE OF A JOB

3.1 Elements of a Job

A "JOB" is the work performed by HSPF in response to a complete set of Users Control Input. It consists of one or more "RUNs" and/or "Time Series Store Management" activities. A RUN is a set of operations which can be performed serially, and which all cover the same period of time (span). The operations are performed in a sequence specified in the Users Control Input. To avoid having to store large quantities of intermediate data on disc, operations may be collected in a group in which they share a common INPAD (INGRP).

3.2 Groups Of Operations

In most runs, time series have to be passed between operations. As described in Section 2.2, each operation can communicate with three different time series storage areas: the TSS, the INPAD, and sequential files. This is illustrated in Fig. 3-1.

Potentially, any time series required by or output by any operation can be stored in the TSS or a sequential file. The user simply specifies the exact origin or destination for the time series, and the HSPF system moves the data between that device and the appropriate row of the INPAD. This system can also be used to transfer data between operations. However, it does require that all transferred data be written to the TSS or a sequential file. This may be very cumbersome and/or inefficient and it is better to transfer data via the INPAD, where possible.

To transfer data via the INPAD, operations must share the same pad. This means that all time series placed in the pad have the same time interval and span. This requirement provides a logical basis for grouping operations; those sharing a common INPAD are called an INGRP (Fig. 3-1). The user specifies the presence of groups in his "Users Control Input (UCI)." A typical sequence of input is shown in Fig. 3-2.

The user also indicates (directly or indirectly) in his control input the source and disposition of all time series required by or output by an operation. If he indicates that a time series must be passed to another operation then the system assumes that the transfer will be made via the scratch pad. If they are not in the same INGRP there is an error. Without a common INPAD, the data must go via the TSS. The structure of the Users Control Input is documented in Part F.

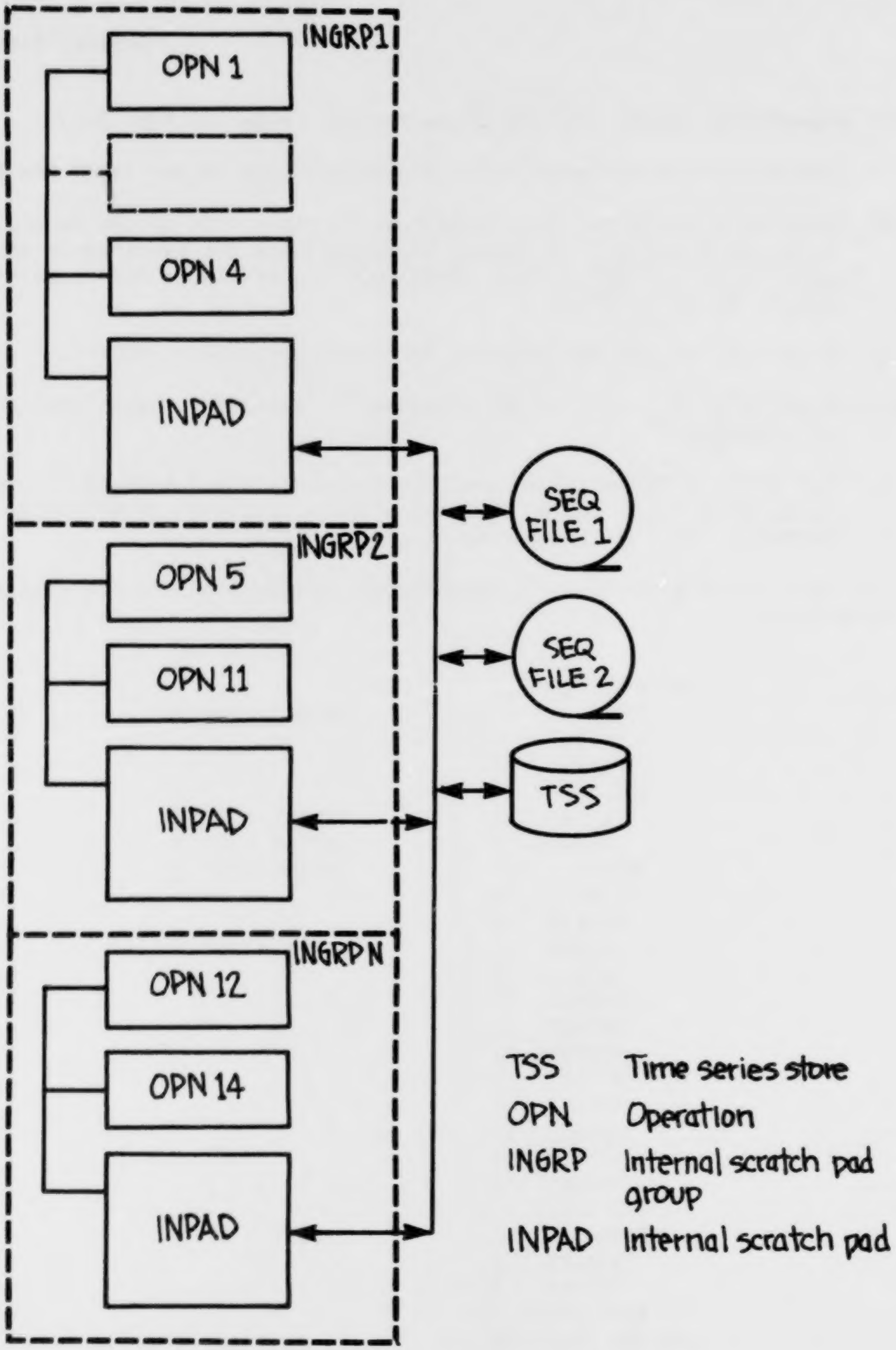


FIG 3-1 SCHEMATIC OF DATA FLOW & STORAGE FOR A SINGLE RUN

The sequence of events in a run is as follows (refer to Fig. 3-1).

(a) Operation 1 is performed until its output rows in the INPAD are filled.

(b) Data are transferred from those rows to other time series storage areas, as required. If any of these data are not required by other operations in INGRP1, their INPAD rows are available for reuse by other operations in INGRP1.

(c) Steps (a) and (b) are repeated for each operation in INGRP1.

(d) Steps (a), (b), and (c) are repeated, if necessary, until the run span is complete.

(e) The INPAD is reconfigured and work on operations 5 through 11 proceeds as in steps (a-d) above. The step repeats until all INGRP's have been handled. The run is now complete.

Note that reconfiguration of a scratch pad implies that its contents will be overwritten.

```
OPN SEQUENCE
    INGRP                  INDELT = 00:30
      COPY     1
      PERLND   1
    END INGRP
      PERLND   2           INDELT = 00:30
      PERLND   3           INDELT = 00:20
    INGRP                  INDELT = 00:30
      COPY     2
      RCHRES   1
      RCHRES   3
      RCHRES   5
      RCHRES  20
      RCHRES  22
      RCHRES  23
      RCHRES   7
      RCHRES   8
      RCHRES  50
      RCHRES 100
      RCHRES 200
    END INGRP
    INGRP                  INDELT = 00:10
      ANALYS   1
      PLTGEN   1
    END INGRP
END OPN SEQUENCE
```

Fig. 3-2 Extract from typical Users Control Input, showing how grouping of operations is specified

PART C STANDARDS AND CONVENTIONS

CONTENTS

Page

1. Structured Programming Technology 24
   1.1 Introduction 24
   1.2 Structure Charts 25
   1.3 HIPO Diagrams 27
   1.4 Pseudo Code 27

2. Terminology 29

3. Fortran Language 30
   3.1 Use of ANSI Fortran 30
   3.2 Allowable Extensions to ANSI Fortran 30

4. Fortran Conventions 31
   4.1 The Need for Conventions 31
   4.2 Conventions used to Implement the Standard Structure Figures . 31
   4.3 Other Conventions 34

5. Conventions Used in Functional Description 42

6. Method of Documenting Data Structures 42
   6.1 Structure of Data in Core 42
   6.2 Structure of Data on Disc Files 42

7. Method of Handling Diagnostic Messages 43

8. References 44

1.0 STRUCTURED PROGRAMMING TECHNOLOGY

1.1 Introduction

Developing, modifying, or even trying to understand, a large computer program can be a very frustrating activity, as many people who have been involved in those kinds of work can testify. Typical problems encountered are:

(1) It is hard to see the relationship between various parts of the program. The underlying design, if there was any, has become obscured as the program evolved to its current state.

(2) The logic of the program is very contorted. The flow chart resembles a bowl of spaghetti.

(3) "Bugs" are difficult to locate.

(4) A seemingly trivial alteration to the program sets off a chain reaction. The resulting problems take a long time to solve.

Why do these things happen? Must long programs necessarily suffer from these ills?

There is no quick answer which covers all aspects of the problem. However, many of these troubles are directly attributable to program structure, or the lack of it. Typically, when the program was designed, an unsuitable structure was chosen or no thought was given to the matter. The reason for this situation is not hard to find. Most engineers are not taught how to design programs; they are only taught how to write program code. There is a great difference between these two activities. It is like the difference between writing a novel and supplying short answers to a series of questions.

The importance of structure emerged as computer scientists investigated the reasons for these problems. As this work went on, a set of new disciplines evolved, which emphasized structure in the design, implementation, and documentation of software. Collectively, they are referred to as "Structured Programming Technology." The individual techniques are:

(1) Structured Design
(2) Top Down Programming
(3) Structured Programming
(4) Hierarchy plus Input, Process, Output (HIPO) Diagrams
(5) use of a Development Support Library

It has been found that the use of these methods, either singly or in combination, leads to computer programs and documentation which are easier to understand than those produced using conventional methods. Also, the software has a unified underlying structure and it is easier to maintain.

Because of its obvious suitability, we made extensive use of Structured Programming Technology on this project. To familiarize us with the techniques, we acquired the "Textbook and Workbook on Structured Programming," and "HIPO - a Design Aid and Documentation Technique" (IBM 1974). These books contain an independent study program which gives the reader a thorough grounding in the underlying principles and practical application of the techniques. Because the Textbook, Workbook, and HIPO Manual are comprehensive and readable, we used them as reference works and kept as close as possible to the concepts and terminology which they contain. This should become apparent on reading further through this document. Through the remainder of this manual, those documents will simply be referred to as the "Textbook," "Workbook," and "HIPO Manual," respectively.

1.2 Structure Charts

A structure chart (SC) is a diagram which shows the functions performed by the various modules in a program and their hierarchy. The use of SC's in Structured Program Design, including the methods used to compile them, is described in Section 6 of the Textbook. We suggest that the reader review that material before proceeding.

The HIPO method of documentation involves the use of a "visual table of contents" (see HIPO Manual), which is very similar to an SC. In our documentation system, we combine these two functions in a single diagram, which we still call an SC, e.g. SC 4.0. The visual table of contents for HSPF is given in Part D of this document.

To compile structure charts, we use a combination of the methods given in the Textbook and in the HIPO Manual, together with some ideas and conventions of our own:

(1) Typically, the SC for a large program extends over several pages. We use the term SC either to describe the entire chart or to refer to that part of it shown on a single page.

(2) Each box in an SC can be viewed in two distinct ways:

- (a) as representing a single subprogram (the one which calls all the subprograms which perform subordinate functions)
- (b) as representing the entire group of subprograms mentioned in (a)

In our documentation system we use this dual meaning to an advantage. Details are given later.

(3) Each box contains a description of the action which it and all its subordinate boxes collectively represent. It follows that the description given in any box must be implied in the descriptions given in all boxes in the chain connecting it to the MAIN program box, e.g. the description for box 4.1.1 in SC 4.1 is implied in (or is part of) the descriptions for boxes 4.1, 4.0, and 1.0. Because it describes the action performed by one or more subprograms, a description starts with the imperative form of a verb, e.g. compute, find, get (HIPO Manual, p. 63). In an SC, the description appears in lower case.

(4) The numbering system is as described in the HIPO Manual. Numbers are placed inside the boxes, near the lower right hand corner. We have introduced one extension to the numbering system because the function "perform an operation" refers to any of the operating modules in the system, which are all conceptually similar and of equal rank. Therefore, we assign them all the number 4.2 and distinguish them only by a subscript. For example, 4.2(3) refers to the Reach/Mixed Reservoir Application Module, and 4.2(3).1 refers to Section 1 of that module (Hydraulics).

(5) There is a name next to each box, outside the top left hand corner. It identifies the Fortran subprogram represented by the box. Note that this name applies to a single box, not an entire subprogram group. It is written in upper case.

(6) An SC bears the number of the highest level box which it depicts, e.g. SC 4.0 shows subprogram 4.0 and its subordinates.

(7) Where an SC continues across more than one page we place "flags" at points where the chart has been broken. They are labeled with the chart number on which the information is continued. For example , see the stubs at the bottom of SC 1.0 in Part D.

The entire set of SCs for the HSPF system appear together in the Visual Table of Contents (Part D). It consists of the full set of Structure Charts, arranged in numerical order. Its purpose is to enable a user to:

(1) see how the following material (the functional description in Part E) is arranged

(2) get an overview of the entire system

(3) locate a given subprogram in the system

(4) see a subprogram in its surroundings, so its relationship to the rest of the system can be better understood

(5) design an overlay structure, if HSPF is being installed on a computer which requires it.

1.3 HIPO Diagrams

A HIPO diagram depicts graphically the function of a subprogram and its subordinate subprograms. A HIPO Diagram bears the same number as the SC box which it illustrates. Thus, for example, HIPO diagram 4.0 describes the functions performed by the entire subprogram group 4 from the perspective of the head subroutine (OSUPER).

Although some HIPO diagrams were prepared when HSPF was designed, we did not find them very useful, so they are not included in this document. As we become more familiar with Structured Program Design, we might develop a better appreciation of their value.

1.4 Pseudo Code

Pseudo code is an English-like noncompilable language which describes program logic. It is discussed in Section 4 of the Textbook. Its chief advantage, compared to Fortran code, is that it encourages the writing of programs with good structure which are easily read and understood. It does not contain statement numbers or GO TO statements. We developed our own version, almost identical to that used in the Textbook. The entire HSPF system was written in pseudo code (Johanson, et al. 1979) before being translated into Fortran. We observed the following conventions:

(1) Only valid structure figures are used. These include the SEQUENCE, IFTHENELSE, WHILEDO, DOUNTIL, and CASE figures.

(2) Successive levels of indentation are displaced from each other by two columns.

(3) Fortran identifiers and subprogram names are in upper case, e.g. RCHTAB.

(4) Other identifiers are in lower case with initial capital letter, e.g. Time-parms. This applies both to names of groups of variables (e.g. Date-time) and to names of any individual variables which have not been reduced to a Fortran identifier, e.g. Land-segment.

(5) Keywords are in upper case, e.g. DOUNTIL, BEGIN, IF, THEN.

(6) Other text is in lower case (no initial capitals).

(7) "CALL" means a subprogram is called.

(8) "INCLUDE" means a named segment of code will be physically inserted.

(9) In a DOUNTIL it is assumed that the initial value of the index (if used) is 1 and that it is incremented by 1, unless other values are previously assigned and/or specified.

(10) Character string constants are enclosed in single quotes, e.g. 'TSSM'.

(11) Two-valued flags have the value ON and OFF or 0 and 1.

(12) Comments are enclosed in brackets.

(13) Where we refer to all the elements of a vector or array we often use the PL/1-like notation involving *. For example,

(a) we use

```
OM(*) = FACT*OS(*)
```

instead of

```
DOUNTIL N = NOFLO
  OM(N) = FACT*OS(M)
ENDDO
```

(b) we use

```
BEGIN SETTPT (RCHTAB(*),----)
```

instead of

```
BEGIN SETTPT(RCHTAB,----)
```

to denote the entire vector RCHTAB

(14) Argument lists are enclosed in parentheses, with input arguments first, input/output arguments next and output arguments last. These groups are separated by 'bb' or placed on separate lines as is done in our Fortran.

(15) We often use a cross reference to save time and trouble, e.g.

```
INCLUDE COMMON/SCRTCH/ version HYDR2
CALL AUXIL (argument list as in AUXIL)
```

(16) We use the following symbols for logical operators:

=, not=, >, <, >=, <=

(17) We follow our Fortran coding conventions, where applicable (Section 4).

(18) Every program unit has a comment immediately following the BEGIN statement, outlining its purpose (as in Fortran).

(19) We kept program units short, as the Textbook recommends. We made free use of "CALL" to achieve this.

(20) Individual statements in pseudo code may correspond to individual Fortran statements or to groups of Fortran statements, for example:

(a)	TABPT = TABPT - 1	(single Ftn stmt)
(b)	Warn that extrapolation will take place	(group of Ftn stmts)

We attempted to make our pseudo code sufficienty detailed that inplementation in Fortran was simple. This helped eliminate errors in the translation process.

2.0 TERMINOLOGY

Successful software development requires both clear thinking and communication of ideas. If a member of a programming team is to function effectively, he must have a clear picture of the problems he is to solve and he must be able to express his solutions precisely. When team members interact they must understand each other clearly. And the final product and its associated documentation must be consistent and unambiguous. This implies that a common set of technical terms and definitions should be used throughout the project.

Unfortunately, in the software field there is considerable confusion over the use of terms. The principal reason is that computer science is still a young discipline. New terms are constantly appearing. The development of Structured Programming Technology has spawned a whole set of jargon. There is also confusion over the meaning of terms which have been in use for some time, e.g. "module" and "segment." Many have poorly defined meanings and, as a result, mean different things to different people. Obviously we need to take the trouble to define our terms.

To minimize this kind of problem we developed a glossary of terms pertaining to HSPF. It is located in Appendix I.

3.0 FORTRAN LANGUAGE

3.1 Use of ANSI Fortran

There are many different "dialects" of the Fortran language. Almost every machine and operating system has its own variation on the basic theme. One of the goals for this project was to produce software which would be portable. That is, it should be capable of running on a wide variety of machines. To do this we had to employ a common subset of the many dialects.

The American National Standards Institute has prepared a specification for Fortran. Most manufacturers now implement all of its provisions, plus their own extensions. We used ANSI Fortran together with a few specified extensions.

3.2 Allowable Extensions to ANSI Fortran

We decided to extend beyond ANSI Fortran because, when applied to our type of work, it does have some deficiencies which prohibit certain actions or, at least, make them awkward. For an extension to be approved it had to meet two criteria:

(1) There had to be a compelling reason for its use in this project.

(2) It had to be a feature incorporated in the Fortran languages of all the following series of machines:
   (a) IBM 360 and 370
   (b) UNIVAC 1100
   (c) HP3000
   (d) PDP11
   We figured that if these four systems all implemented a feature, the chances were good that it was widely available.

The following extensions have been adopted:

(1) Direct access input/output

(2) Literals can be enclosed in single quotes e.g. 'ABC' instead of 3HABC

(3) "Half-word" and "full-word" integers. Although ANSI Fortran does not include two kinds of integers, differing in length, we did make use of this feature. Unfortunately, it is not available in the UNIVAC 1100 (and some other systems).

Adaptation of the code to a system which does not include extension (3) would be quite costly; dispensing with extension (1) would be very costly.

4.0 FORTRAN CONVENTIONS

4.1 The Need for Conventions

In addition to defining a Fortran dialect for this software, we adopted a set of conventions for its use. This was done because conventions promote good programming style and, thus, help to produce consistent code. Usually, there are many many ways of programming the solution to a given problem. Although they may all use the same Fortran dialect and produce the same results, one method may be vastly superior to another because it has a clearer or better structure and implementation. While good computer programming cannot be reduced to a set of rules, the use of appropriate conventions can greatly assist in producing code which has a readily observable and logical structure.

Serious communication problems can arise when several people work on the same set of software. We wanted to produce consistent software, so it was necessary to adopt conventions. However, this did involve a trade-off. Too few conventions would have resulted in code which lacked cohesion; too many would have caused programmers to feel stifled and restricted. We tried to strike a reasonable balance.

4.2 Conventions Used to Implement the Standard Structure Figures

The Fortran language does not easily lend itself to structured programming because:

(1) It is not block structured.

(2) It has an IF statement which allows conditional execution of only a single statement on the true condition.

(3) It has a DO statement which allows for index condition testing only.

Therefore, to write structured programs using Fortran, it is necessary to use programming conventions. In this section the conventions for simulating the following standard structure figures will be outlined:

(1) IFTHENELSE

(2) WHILEDO (called by some DOWHILE)

(3) DOUNTIL

(4) CASE

Because of the lack of block structure, it is necessary to use statement labels in the simulation of the standard figures. These labels are given as lowercase letters in the examples below.

4.2.1 IFTHENELSE

The IFTHENELSE figure tests a single predicate "p" to determine which of the two function blocks (of code) F1 or F2 will be performed. The convention for implementing the IFTHENELSE is:

```
    IF (NOT.p) GO TO a
      Code for F1
      GO TO b
  a CONTINUE
      Code for F2
  b CONTINUE
```

If the ELSE is not used, the figure reduces to the following:

```
    IF (NOT.p) GO TO a
      Code for F1
  a CONTINUE
```

Statements within the clauses are indented two columns. The CONTINUE statements terminate each clause and are coded in line with the IF. The following may be used if the code for F1 is a single statement:

```
    IF (p) Code for F1
```

4.2.2 WHILEDO

We call this figure WHILEDO and not DOWHILE (as in the IBM Manual) because the name reminds one that the test is made before the block is "done." The block will not be executed at all if the condition is initially false.

The WHILEDO is coded as follows:

```
C WHILEDO (optional additional explanatory comment)
a IF (.NOT.(p)) GO TO b
    Code for F
    GO TO a
b CONTINUE
```

The negation of the predicate "p" is used for clarity. In actual fact the condition is generally given without the NOT. Statements within the figure are indented two columns. The CONTINUE clause terminates the figure and is coded in line with the IF.

4.2.3 DOUNTIL

The DOUNTIL figure provides essentially the same loop capability as the WHILEDO, differing from it in two respects:

(a) The test of the predicate p is reversed. The WHILEDO terminates when p is false; the DOUNTIL terminates when p is true.

(b) p is tested after each execution of F so that F is always executed at least once.

This figure can be implemented in two ways:

Method 1

```
C  DOUNTIL (optional explanatory comment)
a  CONTINUE
     Code for F
   IF (.NOT.(p)) GO TO a
```

Statements within the figure are indented two columns. This method is used if an index variable is not needed or if the predicate p is complex.

Method 2

The standard Fortran DO is really a DOUNTIL based on an index. This form has a cleaner appearance and will probably execute faster. It should probably be used even if the index is only a counter.

```
   DO a (index specification)
     Code for F
a  CONTINUE
```

Statements within the figure are indented two columns with respect to the DO and CONTINUE. Note that, in terms of ANSI Fortran, the value of the index is undefined after completion of the loop.

4.2.4 CASE

The CASE form is used to select one of a set of functions to be performed depending on the value of a variable aa with m possible values. The figure is implemented as:

```
C  CASENTRY I
k  GO TO(a, b, ..., n), I
a  CONTINUE
     Code for case 1
     GO TO t
b  CONTINUE
     Code for case 2
     GO TO t
       .
       .
       .
n  CONTINUE
     Code for case n
t  CONTINUE
```

If it is possible for I to acquire an invalid value, the following code can be added to check for that condition:

```
   IF (I.GE.1.AND.I.LE.m) GO TO k
C  Invalid value of I
   code to handle invalid value goes here
```

Statements within each case are indented two columns. The CONTINUE statements separate each case and are coded in line with the computed GO TO.

4.3 Other Conventions

4.3.1 Layout of program units

The Fortran language makes certain stipulations as to the order of statements. We expanded on this, to improve the readability, clarity, and consistency of our program units. The order of things is:

(1) A comment card containing the number of the program unit.

(2) The subprogram header line, including the list of dummy arguments.

(3) A comment of 1-10 lines which describes the general purpose of the program unit.

(4) Type statements for the dummy arguments, if this is a subprogram.

(5) Any COMMON statements and their associated type and EQUIVALENCE statements. The COMMON blocks are all named and occur in alphabetical order.
e.g.
```
COMMON/IO/INP,OUT,ERRMESS
INTEGER INP,OUT
REAL ERRMESS
COMMON/PARMS/A1,ALTER,COAT
INTEGER COAT
REAL A1,ALTER
```

(6) Local variable declarations and any associated EQUIVALENCE statements.

(7) DATA statements, for local arrays and variables

(8) Statement function definitions

(9) FORMAT statements, ordered by statement number. Numbers of READ formats start at 1000 and increase by 10; numbers of WRITE formats start at 2000 and increase by 10.

(10) The executable code. This consists of the standard structure figures and code segments implemented in Fortran using the conventions described earlier.

4.3.2 Control structure of a program unit

Code segments may be composed of any of the Fortran statements. However, the use of GOTO, IF, and DO statements is restricted to the standard structure figures. Control flow enters a code segment at the top and leaves at the bottom. No other entry points or exit points are permitted.

4.3.3 Type statements

The types and dimensions of all variables and arrays used in a program unit are declared; none are typed by default. This is done using the following statements, in the order given:

LOGICAL

INTEGER*2

INTEGER*4

REAL

DOUBLE PRECISION

Hollerith (character) data are stored as REAL variables with four characters per storage location or as INTEGER*2 variables with two characters per storage location. Note that this is designed to fit those machines on which REAL variables occupy four bytes per storage location and integers normally occupy 2 bytes. It is, to some extent, a machine dependent aspect of the software.

The DIMENSION and IMPLICIT statements are prohibited, and dimensioning information is never put in a COMMON statement. The identifiers in each type statement are arranged in alphabetical order, except those which apply to variables in COMMON, which are arranged in their physical sequence.

4.3.4 DATA statements

The DATA statement is used to assign initial values to a set of variables. If it is used to initialize local variables in a subprogram, the compiler ensures that this is done at the start of the run, not each time the subprogram is called. This operation effectively makes the variable "permanent." In a "stack" machine, such as the HP3000, this means that the affected data items occupy space in the data segment throughout the execution of the program, not just when the subprogram is executing. To minimize the loss of working space in the data segment, we avoided the DATA statement as far as possible.

4.3.5 EQUIVALENCE statements

Equivalence statements are used where it is necessary to do something esoteric, such as overlaying sets of data which are not required by the program at the same time, or the assignment of a single name to a group of variables, which is done often in our data structures (Johanson, et al. 1979). The Chief Programmers regulated this type of work.

4.3.6 Identifiers

An identifier is the name of an array, variable, or subprogram. One of the principles of Structured Programming is that "intelligent" identifiers be used. That is, the characters in the identifier should convey to the reader something about the data which they represent. In a large program there are many entities which have to be named, preferably in such a way that the relationships between them are indicated. For example, in a reach we might have water, sediment size 1, sediment size 2, nitrate, DO, etc. Furthermore, we might need to distinguish between the inflowing, stored, and outflowing quantities of these constituents or we might need identifiers which indicate the accumulated outflows of these constituents over the current day, month, and year. Choosing identifiers for hundreds of entities, such that the names are descriptive and fit into a systematic framework, is not a simple task. Therefore, the assignment of these identifiers did not proceed on an ad hoc basis, but a system was devised for naming the variables in each major module. The variables used by each application module are listed in the Programmer's Supplement (Johanson, et al. 1979).

4.3.7 Labeling of Statements

We label only those statements which need them. This improves the efficiency of the object code. Executable statements are labeled in ascending order, starting with 10 and increasing by 10, to allow later insertion of additional labels without upsetting the order. Format statements are collected at the head of the program unit. "Read" formats come first starting at label 1000 and increasing by 10. "Write" formats come next, starting at 2000 and increasing by 10.

Statement labels start in column 1 and are justified on the left.

4.3.8 Layout of executable statements

The executable parts of a program unit consist of the standard structure figures arranged in a hierarchy which expresses the logic of the program. The level in the hierarchy of any piece of code is reflected in its level of indentation from the left margin. Long, complicated statements are avoided.

4.3.9 Continuation of a statement to more than one line

Because of the large number of continuation lines permitted by most compilers, the continuation indicator is used to indicate the order of these lines. We number them 1, 2, 3, etc. For declaration, CALL or subprogram header statements, the final character on a continued line is a comma, to eliminate doubt should the following line be missing. Similarly, for assignment statements, the last character on a continued line is an operator. Never is an identifier or constant broken at a line boundary. The text on a continued line commences at the same level of indentation as the previous line (apart from the character in column 6, of course).

4.3.10 Mixed mode arithmetic

ANSI Fortran does not permit mixed mode arithmetic. We endeavored to observe this requirement and use the "intrinsic" functions (see ANSI Manual) to convert data to a uniform type where necessary. Mixed mode assignment statements are permissible (see ANSI Manual).

4.3.11 Range and precision of numbers

In Fortran there are two types of numbers; INTEGER and REAL. Because these numbers are represented in any machine by a finite number of bits, there is a limit to the range of values which can be represented. With real numbers there is also a limit to the precision with which numbers can be represented. If a situation is encountered where a REAL variable cannot represent a number with sufficient precision, the DOUBLE PRECISION option can be used. The ANSI makes no stipulations as to the range and precision of variables used in Fortran; these factors vary from one type of machine to another. We designed our program so that range and precision capabilities of all of the following computer systems are met:

IBM 360 and 370

UNIVAC 1100

PDP 11

HP3000/II

Appropriate specifications are:

Type	Range		Precision
	From	To	(decimal places)
DOUBLE PRECISION	10**-38	10**38	16.5
REAL	10**-38	10**38	6.5
INTEGER(2 byte)	-32767	32767	exact

Note: In the case of DOUBLE PRECISION and REAL variables, negative and positive ranges are the same. Only the positive range is indicated above.

4.3.12 Use of comments

Comments are used to:

(1) Describe the purpose of a program unit (1-10 lines long).

(2) Separate and describe the function of logical blocks of code (code segments).

(3) Clarify logic, where necessary, and to illuminate subtle points.

They are not used to provide a running commentary on the code or to explain what is already obvious; e.g. C   ASSIGN I TO Z. A well structured program should not need a host of comments.

Comments appear immediately above the code to which they apply. They were written at the same time as the executable code; these are usually better than comments which are inserted afterwards.

Although good comments do nothing for the computer, care was taken over them because they do help the person who has to read the code at a later stage. Composing a good comment also helps the programmer to review the function of that piece of code and, hence, to produce better code.

The text of a comment is indented to the same column as that of the code which surrounds it, so that it does not impair the visual picture of program structure conveyed by indentation.

Examples:

```
    (1) C  *************************
        C
        C  WARNING!  This may not
        C  work on XXXX installation
        C
        C  *************************

    (2)    A = TEMP*A
        C  ACCA will be used in the 'Totals' Block
           ACCA = ACCA + A
```

4.3.13 Transfer of data between program units

In Fortran, data can be transferred between program units in two ways: by using argument lists or COMMON blocks. Argument lists are used whenever feasible because they specify exactly those items of data passed between the calling and the called program units. COMMON blocks, however, tend to grow as software is developed because they usually serve several program units. Typically, there are more data items in these blocks than any given pair of program units requires. Thus, it is difficult to tell exactly which items are being passed between a pair of program units. The development and maintenance of programs becomes more difficult if COMMON blocks are used where argument lists would have sufficed.

A further advantage of argument lists is that they permit arrays in the called subprogram to have variable dimensions. Implementation of this feature is discussed in the following section.

4.3.14 Argument lists

In ANSI Fortran, an item in a list of actual arguments can be any of the following:

(1) a Hollerith constant

(2) a variable name

(3) an array element name

(4) an array name

(5) any other expression

(6) the name of an external procedure

There is little need to use types (5) and (6) and they are avoided where possible.

Arguments can serve either or both of two functions: to transfer information to the called subprogram (input mode) or to retrieve information from it (output mode). Although Fortran does not require that arguments be separated according to their function, we believe it is helpful to do so. We arrange arguments in the following order:

```
        (input,bbinput/output,bboutput)
            or
        (input,bb
         input/output,bb
         output)
```

bb indicates two consecutive blanks.

Examples:

```
       CALL JUNK (JIN1,JIN2,bbFLAG1,FLAG2,bbJOUT1)
                    input      input/output output

       CALL WORKER (bbbbbbbbVALUE)
                             output


       CALL LONG(INP1,INP2,etc          (input)
     2          MOD1,MOD2,etc           (input/output)
     3          OUT1,OUT2,etc)          (output)
```

An argument list can be used to vary the dimensions of arrays in the called subprogram. In ANSI Fortran this is done by including the name of the affected array, together with scalar variables which indicate its dimensions, in the argument list.

Example:

```
      SUBROUTINE DOIT (A,MAXI,MAXJ)
      INTEGER MAXI,MAXJ
      REAL A(MAXI,MAXJ)
      etc.
```

We make use of this feature when we have reason to believe that an array will have its dimensions changed as the model is applied to different watersheds or as it is implemented on different computer installations. In this way only the highest subprogram in which the array occurs will need to be recompiled when the program is reconfigured. This construction is also sometimes used in a subprogram which is called by several other program units. Thus, the dimensions of dummy arrays in the subprogram automatically agree with the dimensions of the real arrays in the calling program units.

4.3.15 COMMON blocks

Only labeled COMMON is used. The COMMON statement contains a list of variable and array names. It does not contain dimension information; this is put in the associated type statement.

Each COMMON block is immediately followed by statements which declare the type and dimensions of all the variables in the block. The layout of the type statements is described in Section 4.3.3.

Example:

```
      COMMON/ABAG/HEAVY,LIGHT,LIGHTR,K1
      INTEGER K1
      REAL LIGHT(20),LIGHTR(10,2)
      DOUBLE PRECISION HEAVY(10)
```

Because COMMON blocks usually serve several program units and because we wanted to control their use carefully, the layout of data in COMMON was supervised by the Chief Programmer.

4.3.16 General comments on programming style

Good programmers write code for the benefit of other people, not just for the computer. They avoid the use of "clever" constructs which, although efficient in execution, are obscure.

In general, if we were faced with a conflict between machine efficiency and clarity of the code, our chosen solution favored the latter goal.

5.0 CONVENTIONS USED IN FUNCTIONAL DESCRIPTION

The primary purpose of the Functional Description (Part E) is:

(1) to describe the functions performed by the various subprograms in more detail than can be achieved in the Structure Charts

(2) to explain the technical algorithms and equations which the code implements.

Subprograms are described in numerical order in the text. This system provides a logical progression for the descriptions. General comments regarding a group of subprograms can be made when the "top" subprogram is described, while details specific to an individual subordinate subprogram can be deferred until that part is described. For example, a general description of the PERLND module (Section 4.2(1)) is followed by more detailed descriptions of its twelve sections, ATEMP (Section 4.2(1).1) through TRACER (Section 4.2(1).12).

6.0 METHOD OF DOCUMENTING DATA STRUCTURES

6.1 Structure of Data in Core

The way in which we arrange the variables used in our programs is important. We structure them, as far as possible, using techniques like those used in Structured Program Design. We try to group data items that logically belong together.

Most of the variables in an Operating Module are contained in the Operation Status Vector (OSV). The OSVs for the application modules are shown in the Programmer's Supplement (Johanson, et al. 1979). The format used to document a data structure is similar to that used to declare a "structure" in PL/1. We do this because the technique is logical and convenient, not because of language considerations.

6.2 Structure of Data on Disk Files

The HSPF system makes use of several different classes of disk-based data files:

(1) The Time Series Store (TSS) is described in Section 2.0, Part E and in Appendix III.

(2) The instruction files (OSUPFL, TSGETF, TSPUTF) and the OSVFL are documented in the Programmer's Supplement.

(3) The information file (INFOFL), error message file (ERRFL) and warning message file (WARNFL) are self documenting. One need only list the file and read it to understand its contents.

7.0 METHOD OF HANDLING DIAGNOSTIC MESSAGES

HSPF makes use of two kinds of diagnostic message; error messages and warnings. These messages are all stored on two files; ERRFL and WARNFL. This system has at least two advantages:

(1) Because the messages are not embedded in the Fortran, they do not normally occupy any core storage. This reduces the length of the object code.

(2) The files are self documenting. They contain not only all the messages, but other explanatory material. A user need only obtain a line printer listing of the files to get an up-to-date copy of this documentation.

Each message has been given a "maximum count". If the count for a message reaches this value, HSPF informs the user of the fact. Then:

(1) If it is an error message, HSPF quits.

(2) If it is a warning, HSPF continues but suppresses any future printing of this message.

In addition to the above features, the Run Interpreter has been designed to:

(1) Stop if 20 errors of any kind have been detected. This gives the user a fair number of messages to work on, but avoids producing huge quantities of error messages, many of which may be spurious (say, if the code could not recover from early error conditions).

(2) Stop at the end of its work if any errors have been detected by it. Thus, HSPF will not enter any costly time loop if the Run Interpreter has found any errors in the User's Control Input.

8.0 REFERENCES

International Business Machines Inc. 1974. Structured Programming Textbook & Workbook — Independent Study Program.

International Business Machines Inc. 1974. HIPO - A Design Aid and Documentation Technique, Report GC20-1851-1. 130 pp.

American National Standards Institute. 1966. USA Standard Fortran, Standard X3.9-1966. 36 pp.

Johanson, R.C., J.C. Imhoff and H.H. Davis, Jr. 1979. Programmer's Supplement for the Hydrological Simulation Program - Fortran. This material is on magnetic tape. See Appendix IV.

PART D

VISUAL TABLE OF CONTENTS

General Comments 49

FIGURES

Structure chart no.		Page
1.0	Upper levels of HSPF system	49
1.2	Service subprograms available to the entire HSPF system .	50
2.0	Subroutine group TSSMGR	51
2.001	Service subprograms for subroutine group TSSMGR	52
3.0	The Run Interpreter	53
3.01	Service subprograms for the Run Interpreter	54
3.2	Subroutine group SEQBLK of the Run Interpreter.	55
3.4	Subroutine group OPNBLK of the Run Interpreter.	56
3.4.1	Service subprograms for processing input for operating modules	57
3.4(1)	Processing of input for the PERLND module	58
3.4(1).10	Processing of input for the PEST section of the PERLND module	59
3.4(1).11	Processing of input for the NITR section of the PERLND module	60
3.4(1).12	Processing of input for the PHOS section of the PERLND module	61
3.4(2)	Processing of input for the IMPLND module	62
3.4(3)	Processing of input for the RCHRES module	63
3.4(3).2	Processing of input for the HYDR section of the RCHRES module	64
3.5	Processing of User's Control Input which deals with time series.	65
3.5.01	Service subprograms for the TIMSER subroutine group . . .	66
3.5.2	Processing entries in the EXT SOURCES Block	67
3.5.2.2	Subroutine group EXTTS.	68
3.5.2.3	Check and expand a reference to an Operation time series.	69

3.5.3 Processing entries in the NETWORK block 70
3.5.4 Processing entries in the EXT TARGETS Block 71
3.5.6 Subprograms involved in allocation & deallocation of INPAD rows 72
3.5.8 Subroutine group TINSTR 73
3.5.8.01 Service routines for subroutine group TINSTR. 74
3.5.8.2.3 Subroutine group PINIT. 75
4.0 Operations group of modules 76
4.01 Service subprograms for the Operations Supervisor 77
4.1 TSGET 78
4.1.01 Service routines for TSGET and TSPUT. 79
4.1.1 The GETTSS section of the TSGET module. 80
4.1.2 The GETSEQ section of the TSGET module. 81
4.2(1) Pervious land-segment application module. 82
4.2(1).2 The SNOW section of modules PERLND and IMPLND 33
4.2(1).3 The PWATER section of the PERLND application module . . . 84
4.2(1).4 The SEDMNT section of the PERLND application module . . . 85
4.2(1).7 The PQUAL section of the PERLND application module. . . . 86
4.2(1).8 The MSTLAY section of the PERLND application module . . . 87
4.2(1).9 The PEST section of the PERLND application module 88
4.2(1).10 The NITR section of the PERLND application module 89
4.2(1).11 The PHOS section of the PERLND application module 90
4.2(1).12 The TRACER section of the PERLND application module . . . 91
4.2(1).13 Subroutine group PPTOT. 92
4.2(1).14 Subroutine group PBAROT 93
4.2(1).15 Subroutine group PPRINT 94
4.2(1).15.1 Subroutine group PERACC 95
4.2(1).15.2 Subroutine group PERPRT 96
4.2(1).15.3 Subroutine group PERRST 97
4.2(2) Impervious land-segment application module. 98
4.2(2).3 The IWATER section of the IMPLND application module . . . 99
4.2(2).4 The SOLIDS section of the IMPLND application module . . . 100
4.2(2).6 The IQUAL section of the IMPLND application module. . . . 101
4.2(2).7 Subroutine group IPTOT. 102
4.2(2).8 Subroutine group IBAROT 103
4.2(2).9 Subroutine group IPRINT 104
4.2(2).9.1 Subroutine group IMPACC 105
4.2(2).9.2 Subroutine group IMPPRT 106
4.2(2).9.3 Subroutine group IMPRST 107
4.2(3) Reach/mixed reservoir application module. 108
4.2(3).1 The HYDR section of the RCHRES application module 109
4.2(3).4 The HTRCH section of the RCHRES application module. . . . 110
4.2(3).5 The SED section of the RCHRES application module. 111
4.2(3).7 The RQUAL section of the RCHRES application module. . . . 112
4.2(3).7.1 OXRX subroutine group. DO, BOD simulation 113
4.2(3).7.2 NUTRX subroutine group. Primary inorganic N and P simulation. 114
4.2(3).7.3 Simulate plankton populations and associated reactions. . 115
4.2(3).7.3.3 Simulate phytoplankton. 116
4.2(3).7.3.5 Simulate benthic algae. 117

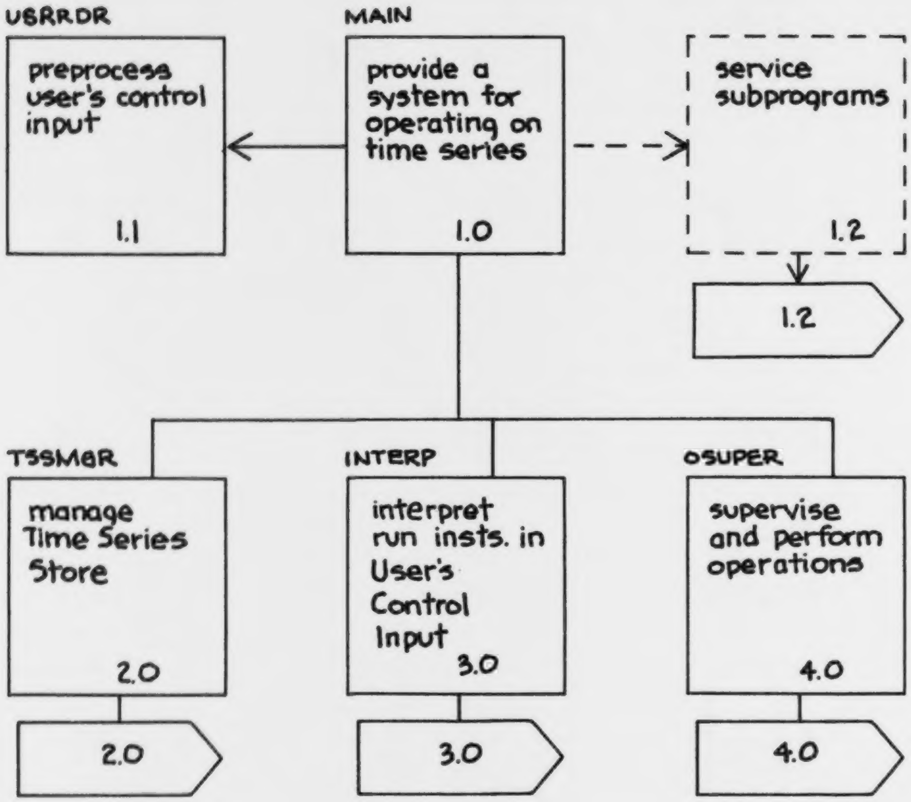
4.2(3).7.4 Simulate pH and carbon species. 118
4.2(3).8 Subroutine group RPTOT. 119
4.2(3).9 Subroutine group RBAROT 120
4.2(3).10 The RPRINT section of the RCHRES application module . . . 121
4.2(3).10.1 Subroutine group RCHACC 122
4.2(3).10.2 Subroutine group RCHPRT 123
4.2(3).10.3 Subroutine group RCHRST 124
4.2(13) The Display utility module. 125
4.2(14) The Duration Analysis utility module. 126
4.3 TSPUT 127
4.3.1 The PUTTSS section of module TSPUT. 128
4.3.1.3 Subroutine group FILTSS 129

GENERAL COMMENTS

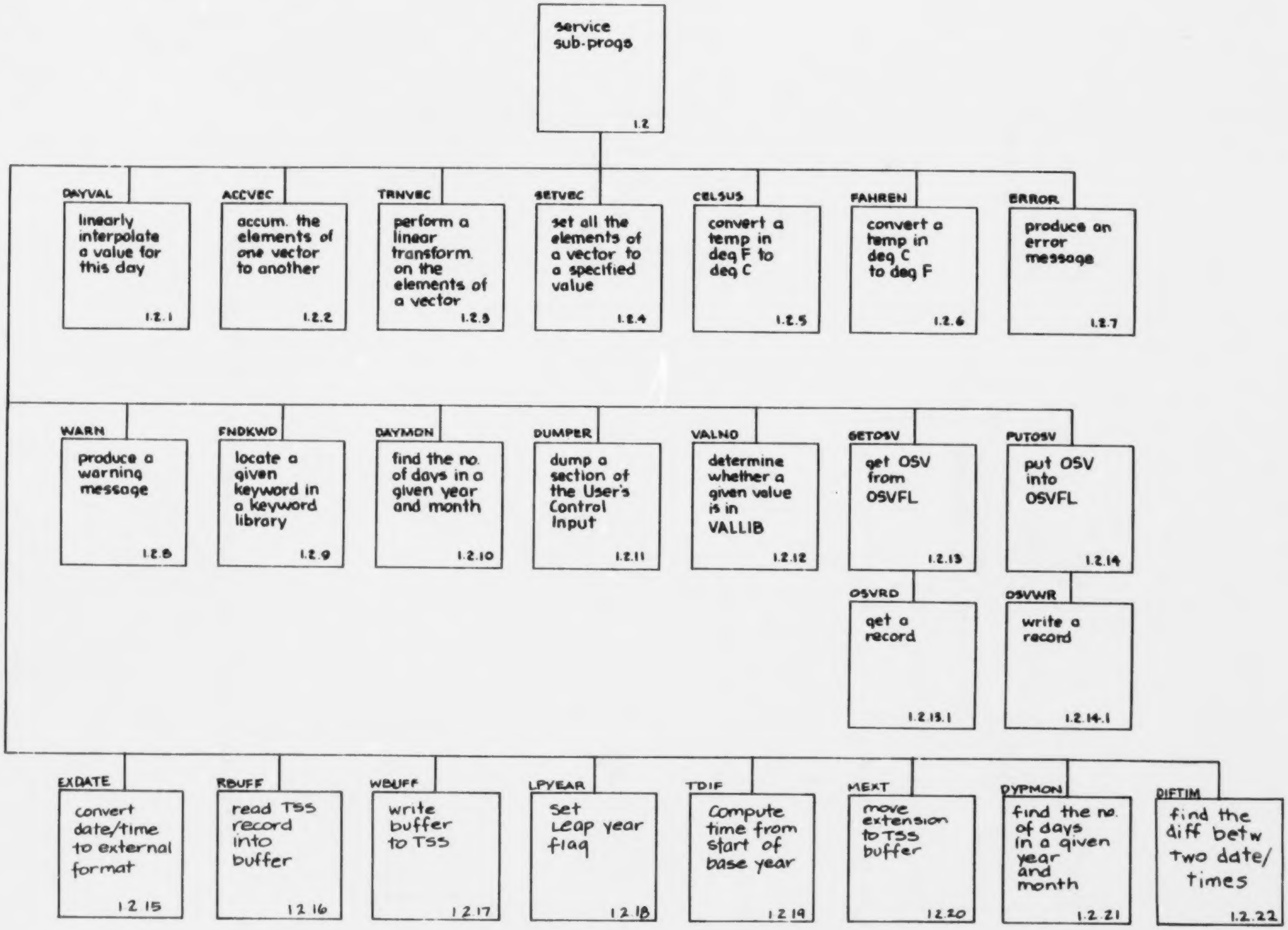
The entire set of structure charts for the HSPF system appears on the immediately following pages, forming a visual table of contents for the software. For a discussion of the conventions followed in compiling the charts, refer to Section 1.2 of Part C "Standards and Conventions".

Note that the following material gives a complete picture of the system. It starts at the highest, most general, level and proceeds down to the lowest, most detailed, levels. A user need not assimilate it all before using HSPF. Initially, one may not wish to proceed more than two or three levels down the structure tree. Later, having become more familiar with the system, one may wish to explore it in more detail.

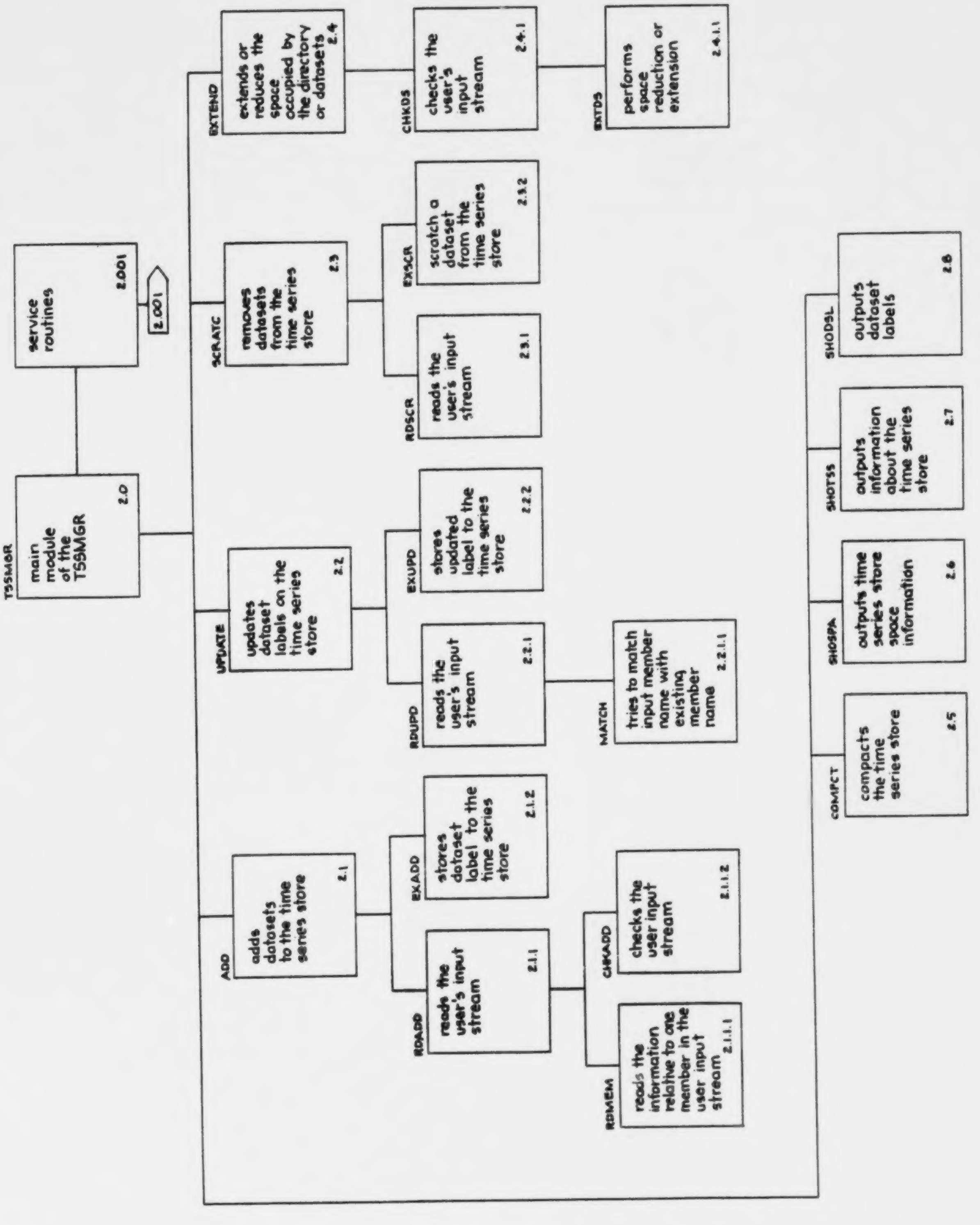
In general, a subprogram calls only those subprograms situated immediately below it. However, "service subprograms" are an exception to this rule. These routines perform elementary tasks, usually for more than one calling subprogram. Therefore, they are arranged in groups; each group is situated immediately above the subprograms which its members can serve. For example, the subprograms in Structure chart 1.2 can be called by any subprogram in HSPF; those in Structure chart 3.01 can be called by any subprogram in the Run Interpreter (Structure chart 3.0). This arrangement makes the structure charts more compact and readable and will assist programmers who need to design overlay structures for the code, or partition it in some other way.


USRRDR
preprocess user's control input
1.1
MAIN
provide a system for operating on time series
1.0
service subprograms
1.2
1.2
TSSMGR
manage Time Series Store
2.0
2.0
INTERP
interpret run insts. in User's Control Input
3.0
3.0
OSUPER
supervise and perform operations
4.0
4.0

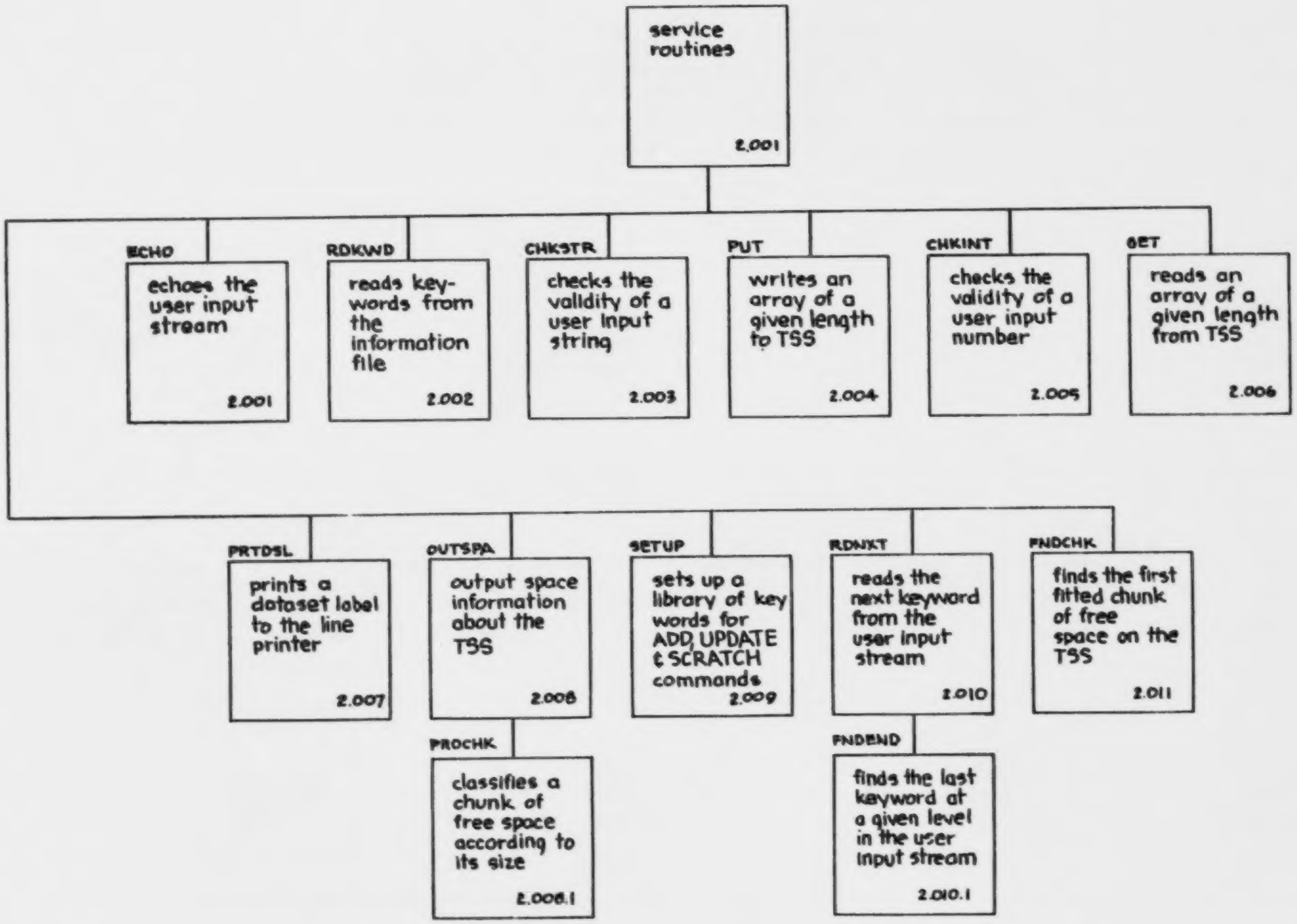
Structure chart 1.0 Upper levels of HSPF system


service sub-progs 1.2
DAYVAL
linearly interpolate a value for this day 1.2.1
ACCVEC
accum. the elements of one vector to another 1.2.2
TRNVEC
perform a linear transform. on the elements of a vector 1.2.3
SETVEC
set all the elements of a vector to a specified value 1.2.4
CELSUS
convert a temp in deg F to deg C 1.2.5
FAHREN
convert a temp in deg C to deg F 1.2.6
ERROR
produce an error message 1.2.7
WARN
produce a warning message 1.2.8
FNDKWD
locate a given keyword in a keyword library 1.2.9
DAYMON
find the no. of days in a given year and month 1.2.10
DUMPER
dump a section of the User's Control Input 1.2.11
VALNO
determine whether a given value is in VALLIB 1.2.12
GETOSV
get OSV from OSVFL 1.2.13
PUTOSV
put OSV into OSVFL 1.2.14
OSVRD
get a record 1.2.13.1
OSVWR
write a record 1.2.14.1
EXDATE
convert date/time to external format 1.2.15
RBUFF
read TSS record into buffer 1.2.16
WBUFF
write buffer to TSS 1.2.17
LPYEAR
set Leap year flag 1.2.18
TDIF
Compute time from start of base year 1.2.19
MEXT
move extension to TSS buffer 1.2.20
DYPMON
find the no. of days in a given year and month 1.2.21
DIFTIM
find the diff betw two date/times 1.2.22

Structure chart 1.2 Service subprograms available to the entire HSPF system

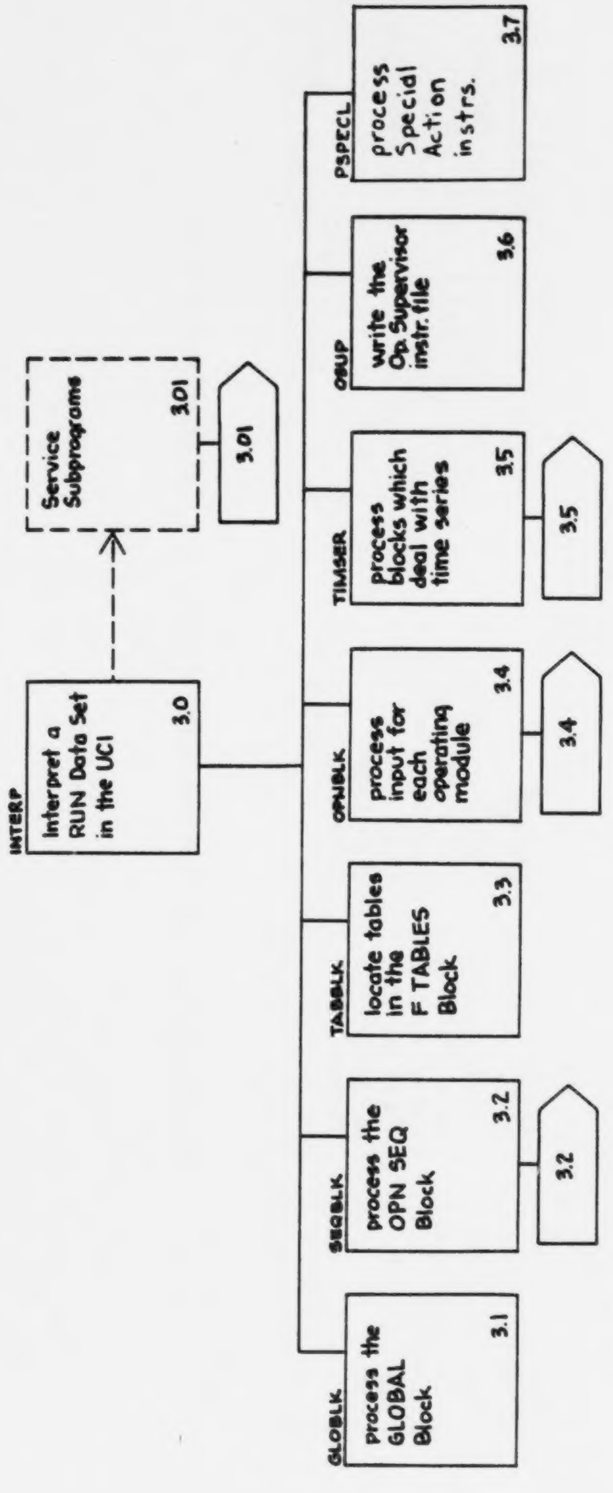

TSSMGR
main module of the TSSMGR
2.0
service routines
2.001
2.001
ADD
adds datasets to the time series store
2.1
RDADD
reads the user's input stream
2.1.1
RDMEM
reads the information relative to one member in the user input stream
2.1.1.1
CHKADD
checks the user input stream
2.1.1.2
EXADD
stores dataset label to the time series store
2.1.2
UPDATE
updates dataset labels on the time series store
2.2
RDUPD
reads the user's input stream
2.2.1
MATCH
tries to match input member name with existing member name
2.2.1.1
EXUPD
stores updated label to the time series store
2.2.2
SCRATC
removes datasets from the time series store
2.3
RDSCR
reads the user's input stream
2.3.1
EXSCR
scratch a dataset from the time series store
2.3.2
EXTEND
extends or reduces the space occupied by the directory or datasets
2.4
CHKDS
checks the user's input stream
2.4.1
EXTDS
performs space reduction or extension
2.4.1.1
COMPCT
compacts the time series store
2.5
SHOSPA
outputs time series store space information
2.6
SHOTSS
outputs information about the time series store
2.7
SHODSL
outputs dataset labels
2.8

Structure chart 20 Subroutine group TSSMGR

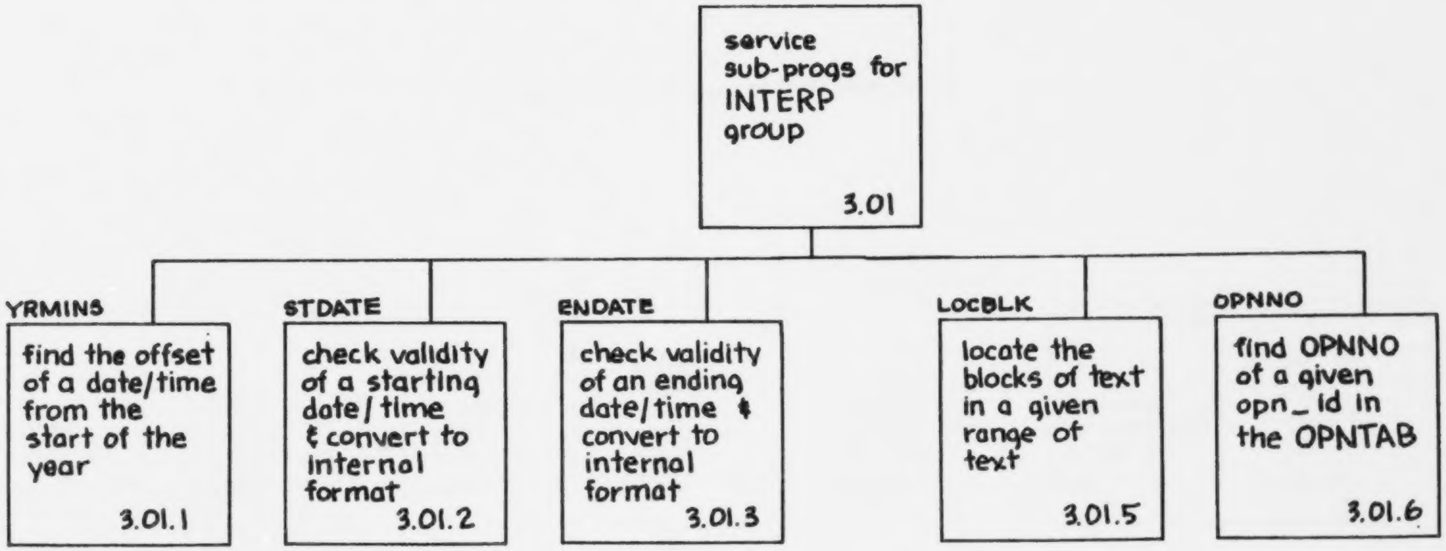


service routines
2.001
ECHO
echoes the user input stream
2.001
RDKWD
reads key-words from the information file
2.002
CHKSTR
checks the validity of a user input string
2.003
PUT
writes an array of a given length to TSS
2.004
CHKINT
checks the validity of a user input number
2.005
GET
reads an array of a given length from TSS
2.006
PRTDSL
prints a dataset label to the line printer
2.007
OUTSPA
output space information about the TSS
2.008
PROCHK
classifies a chunk of free space according to its size
2.008.1
SETUP
sets up a library of key words for ADD, UPDATE & SCRATCH commands
2.009
RDNXT
reads the next keyword from the user input stream
2.010
FNDEND
finds the last keyword at a given level in the user input stream
2.010.1
FNDCHK
finds the first fitted chunk of free space on the TSS
2.011

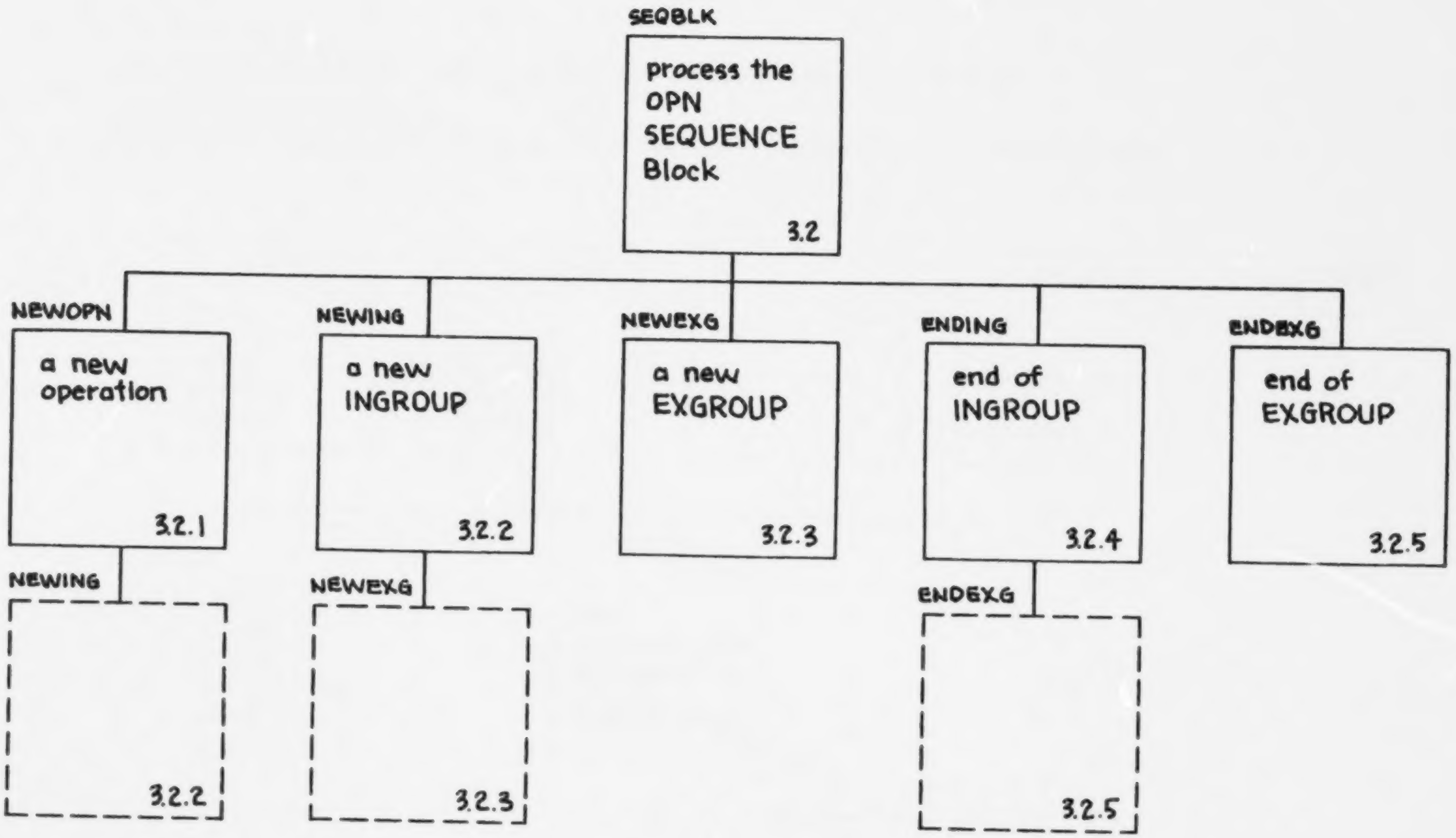
Structure chart 2.001 Service subprograms for subroutine group TSSMGR


INTERP
Interpret a RUN Data Set in the UCI
3.0
Service Subprograms
3.01
3.01
GLOBLK
process the GLOBAL Block
3.1
SEQBLK
process the OPN SEQ Block
3.2
3.2
TABBLK
locate tables in the F TABLES Block
3.3
OPNBLK
process input for each operating module
3.4
3.4
TIMSER
process blocks which deal with time series
3.5
3.5
OSUP
write the Op. Supervisor instr. file
3.6
PSPECL
process Special Action instrs.
3.7

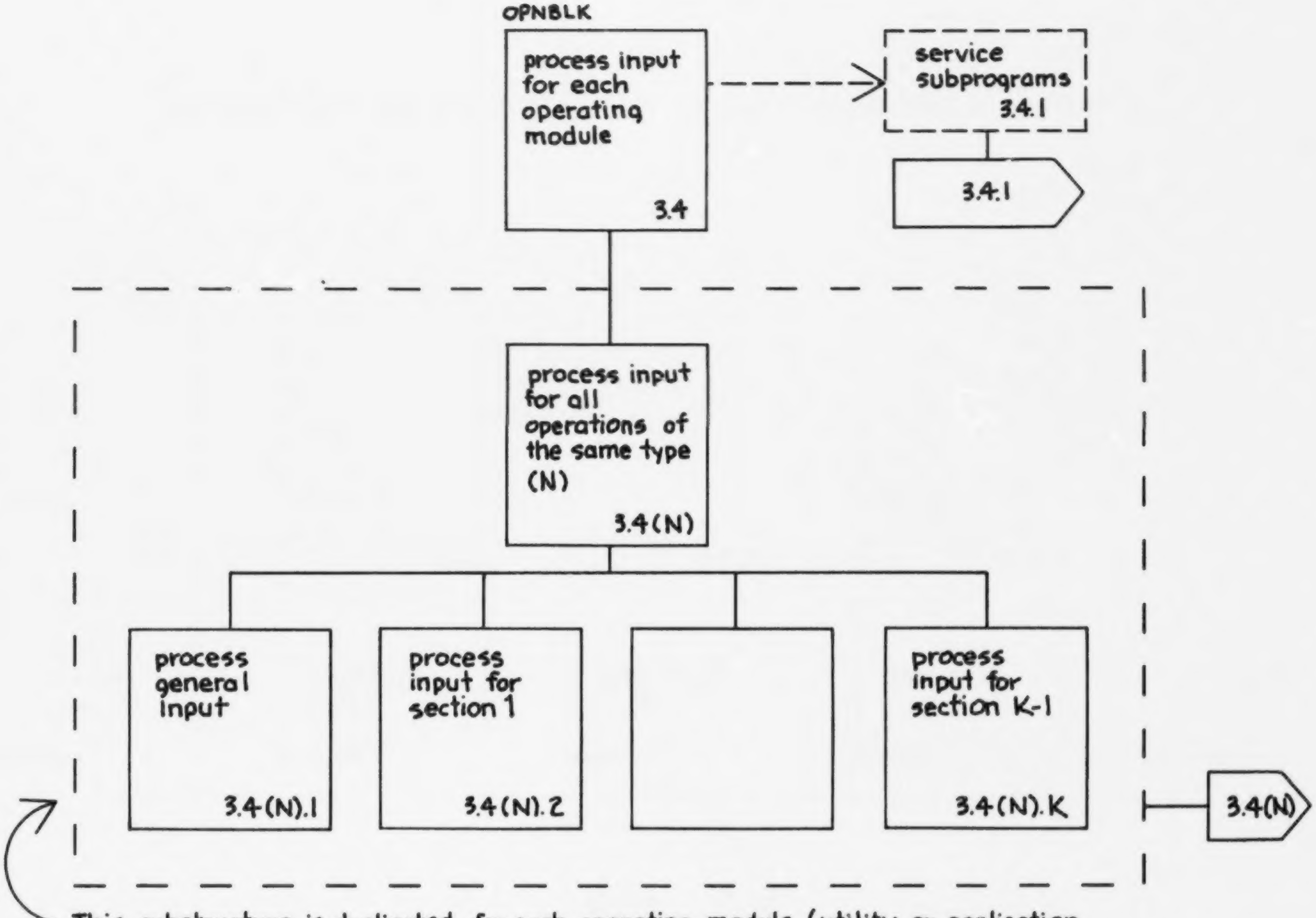
Structure chart 3.0 The Run Interpreter


service sub-progs for INTERP group 3.01
YRMINS
find the offset of a date/time from the start of the year 3.01.1
STDATE
check validity of a starting date/time & convert to internal format 3.01.2
ENDATE
check validity of an ending date/time & convert to internal format 3.01.3
LOCBLK
locate the blocks of text in a given range of text 3.01.5
OPNNO
find OPNNO of a given opn_id in the OPNTAB 3.01.6

Structure chart 3.01 Service subprograms for the Run Interpreter



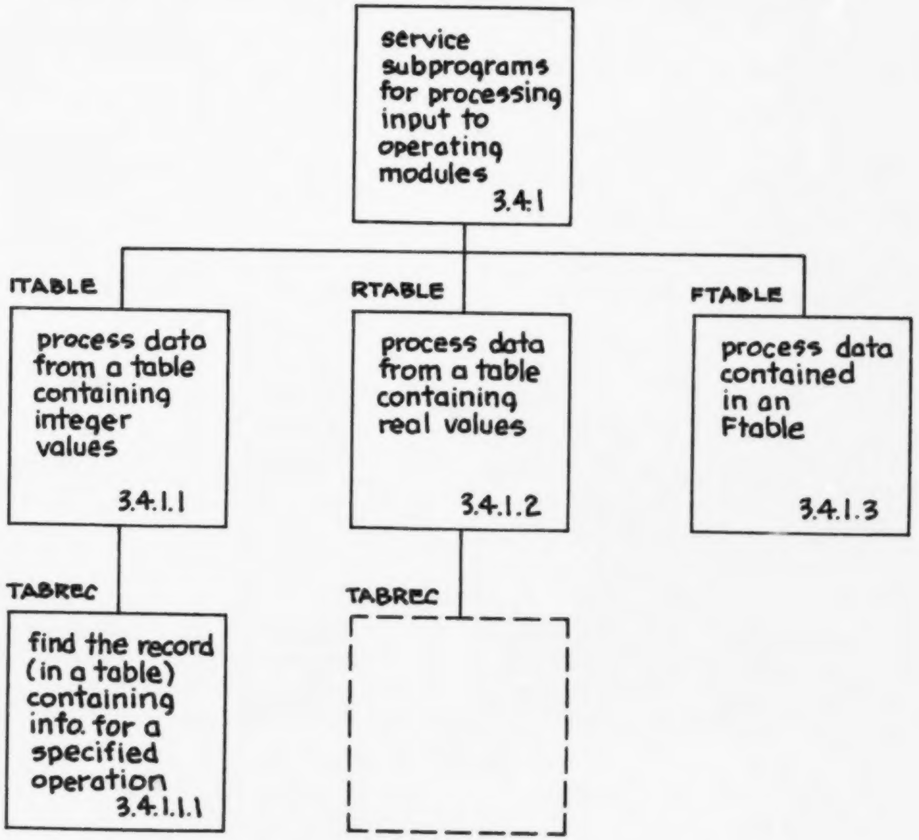
Structure chart 3.2 Subroutine group SEQBLK of the Run Interpreter


OPNBLK
process input for each operating module
3.4
service subprograms
3.4.1
3.4.1
process input for all operations of the same type (N)
3.4(N)
process general input
3.4(N).1
process input for section 1
3.4(N).2
process input for section K-1
3.4(N).K
3.4(N)

This substructure is duplicated for each operating module (utility or application module) in the system.

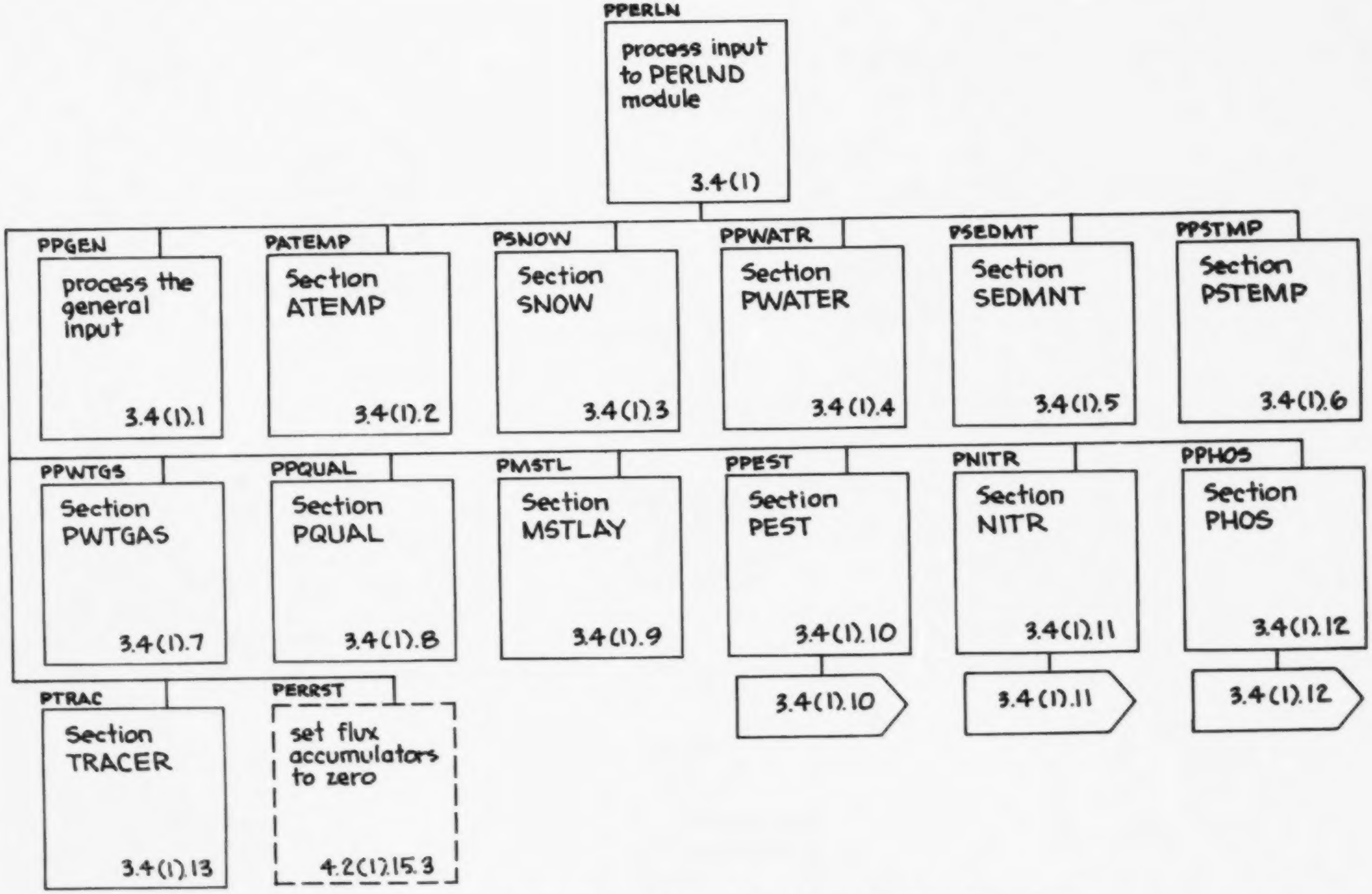
The numbering system shown is that for the Nth operating module.

Structure chart 3.4 Subroutine group OPNBLK of the RUN Interpreter

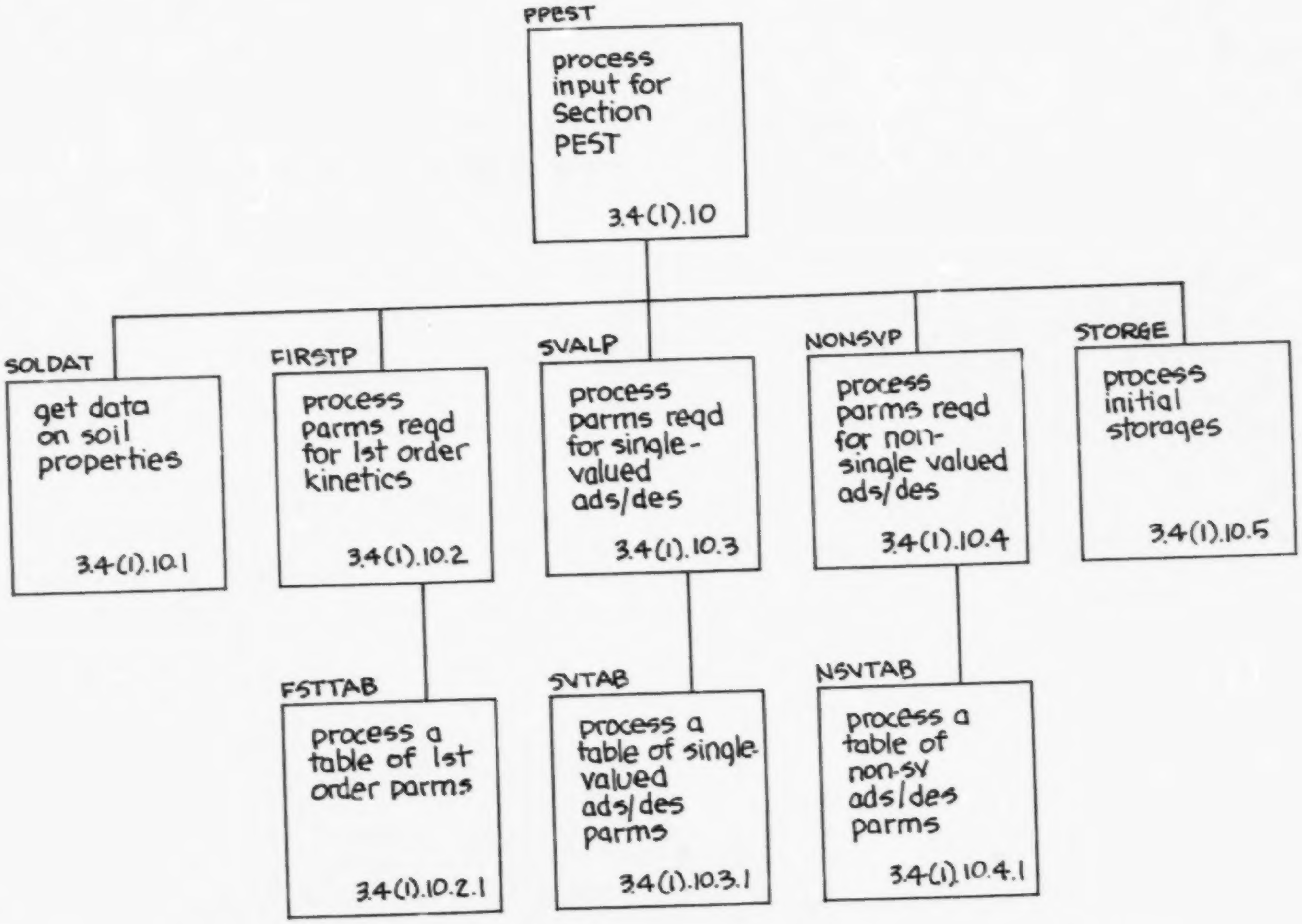


service subprograms for processing input to operating modules
3.4.1
ITABLE
RTABLE
FTABLE
process data from a table containing integer values
3.4.1.1
process data from a table containing real values
3.4.1.2
process data contained in an Ftable
3.4.1.3
TABREC
TABREC
find the record (in a table) containing info. for a specified operation
3.4.1.1.1

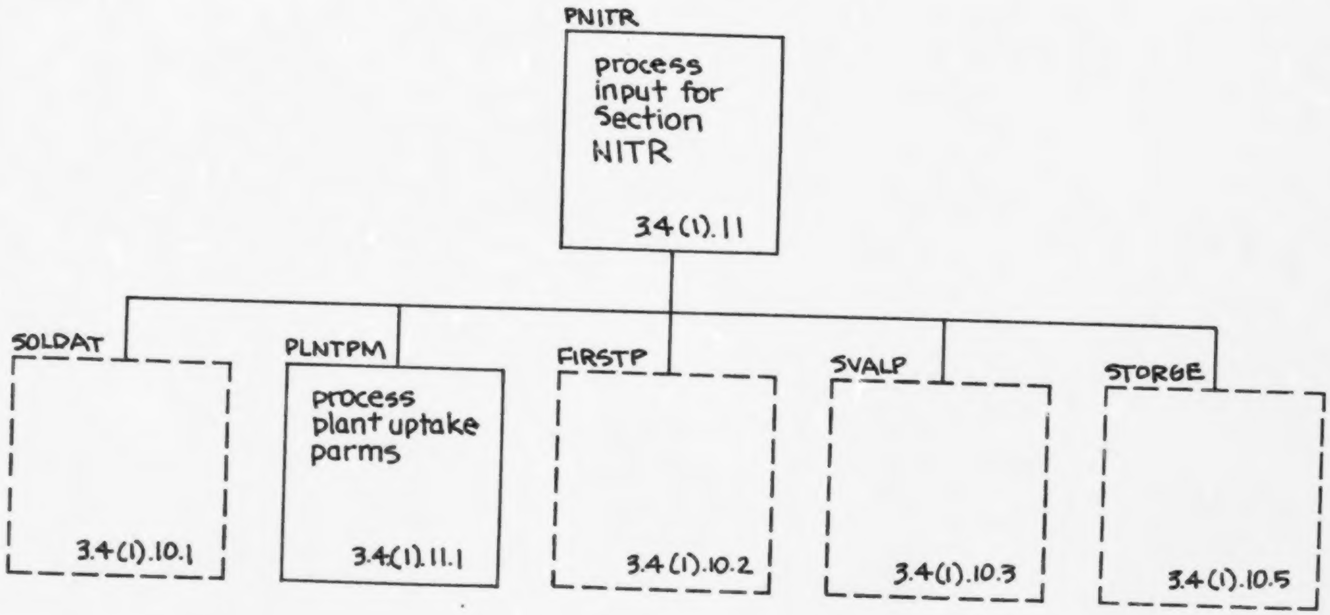
Structure chart 3.4.1 Service subprograms for processing input for operating modules.



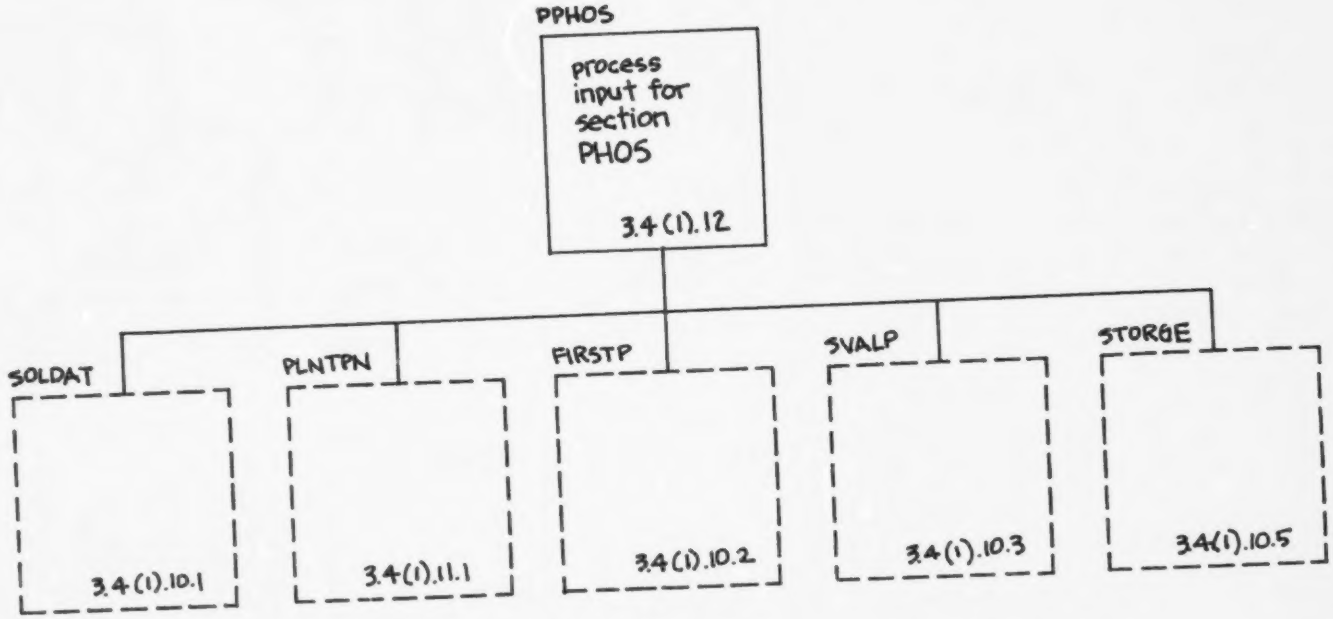
Structure chart 3.4(1). Processing of input for the PERLND module


PPEST
process input for Section PEST
3.4(1).10
SOLDAT
get data on soil properties
3.4(1).10.1
FIRSTP
process parms reqd for 1st order kinetics
3.4(1).10.2
SVALP
process parms reqd for single-valued ads/des
3.4(1).10.3
NONSVP
process parms reqd for non-single valued ads/des
3.4(1).10.4
STORGE
process initial storages
3.4(1).10.5
FSTTAB
process a table of 1st order parms
3.4(1).10.2.1
SVTAB
process a table of single-valued ads/des parms
3.4(1).10.3.1
NSVTAB
process a table of non-sv ads/des parms
3.4(1).10.4.1

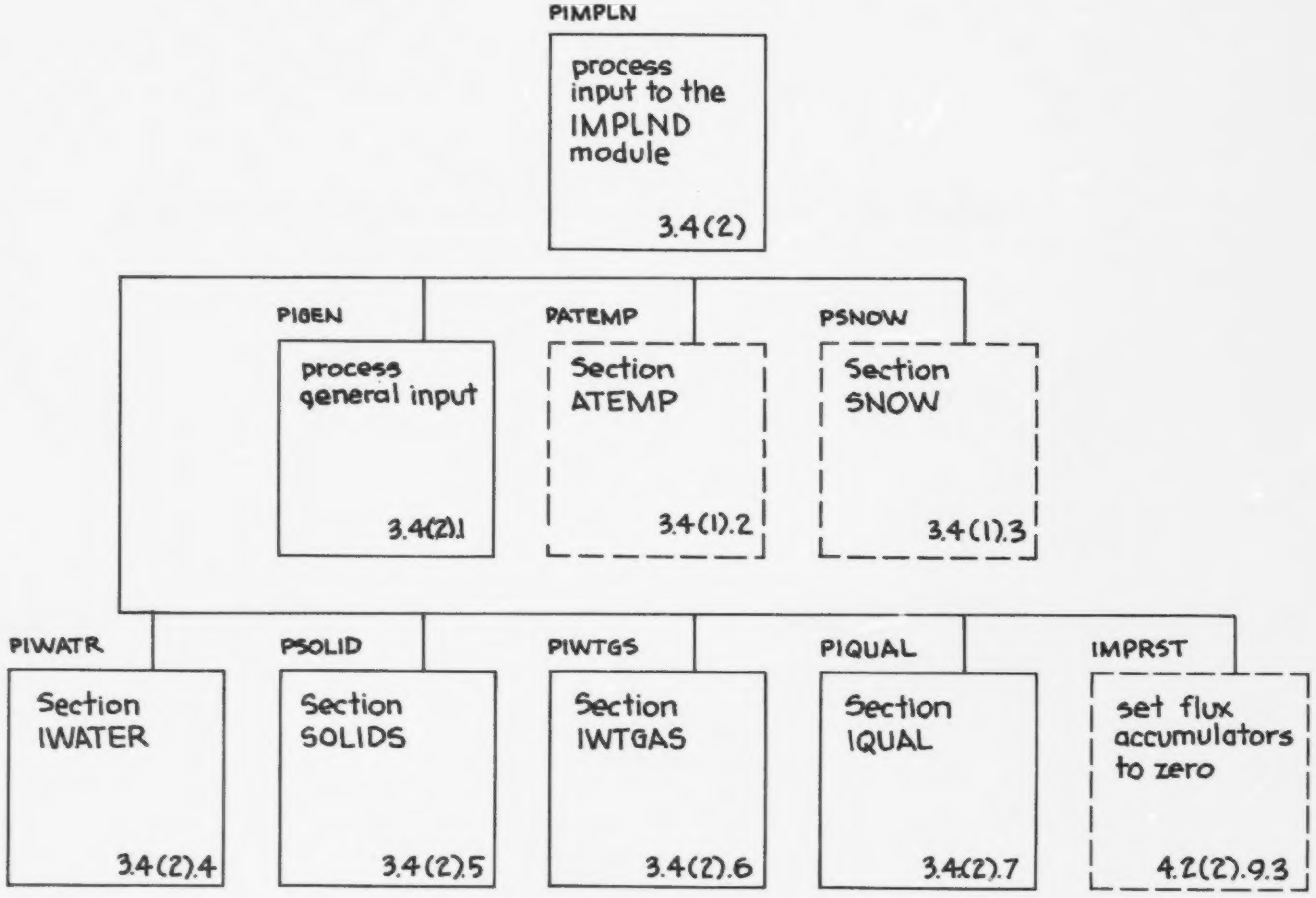
Structure chart 3.4(1).10 Processing of input for the PEST section of the PERLND module



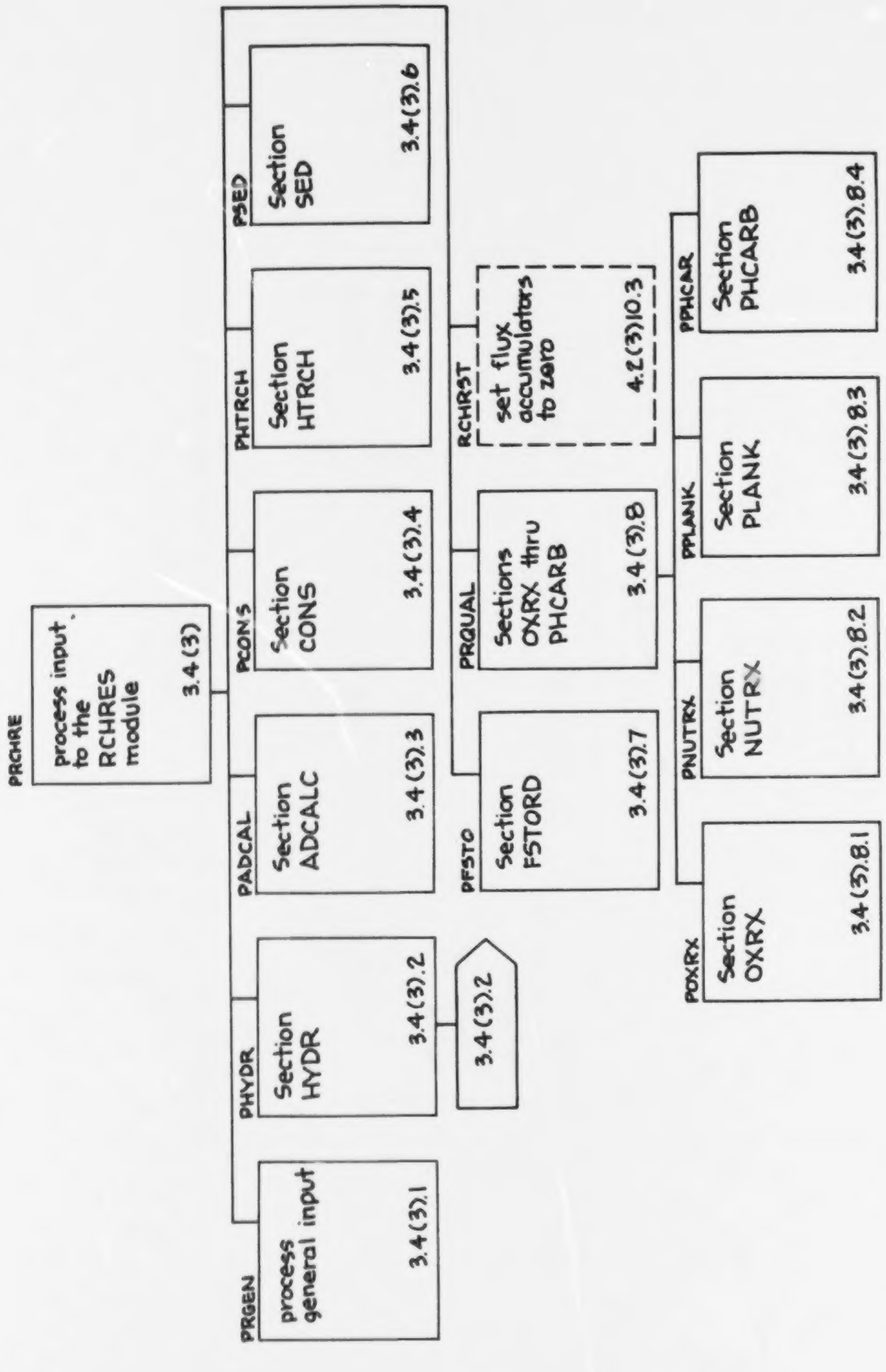
Structure chart 3.4(1).11 Processing of input for the NITR section of module PERLND



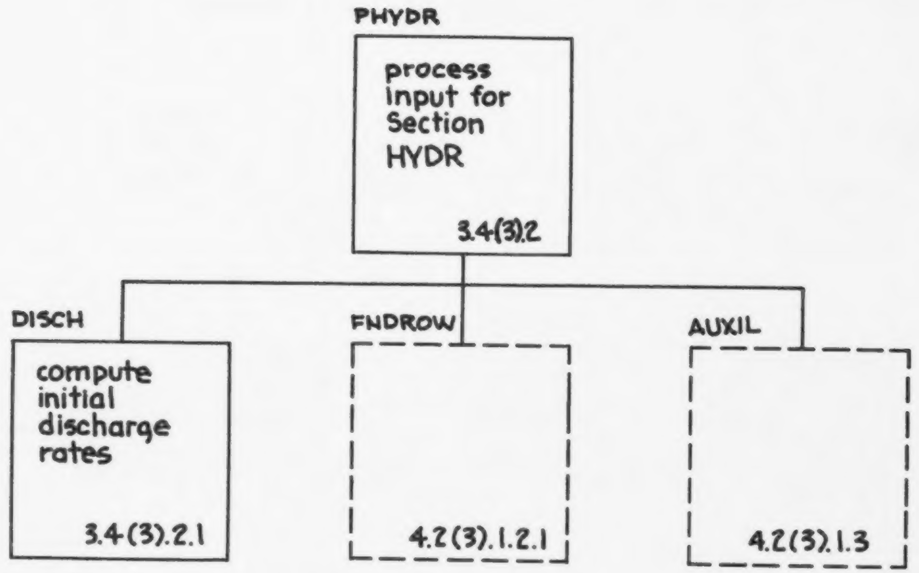
Structure chart 3.4(1).12 Processing of input for the PHOS section of module PERLND



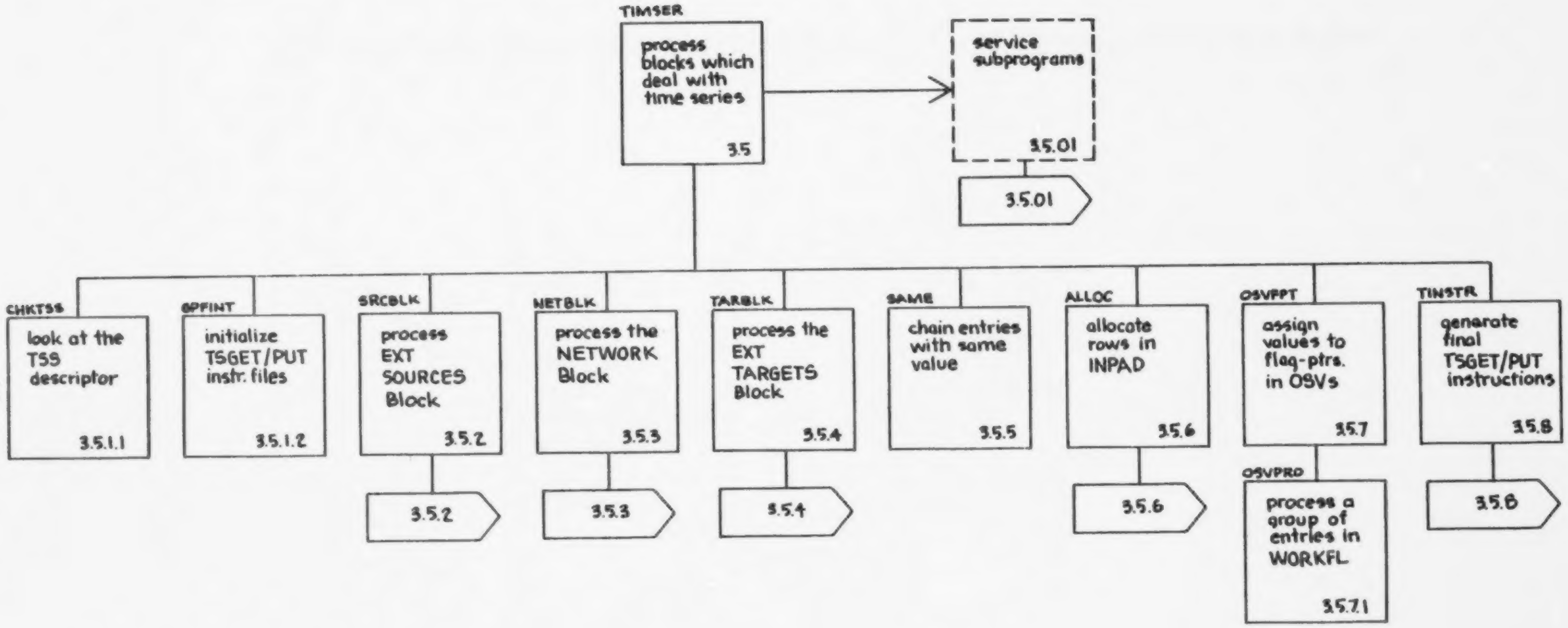
Structure chart 3.4(2) Processing of input for the IMPLND module



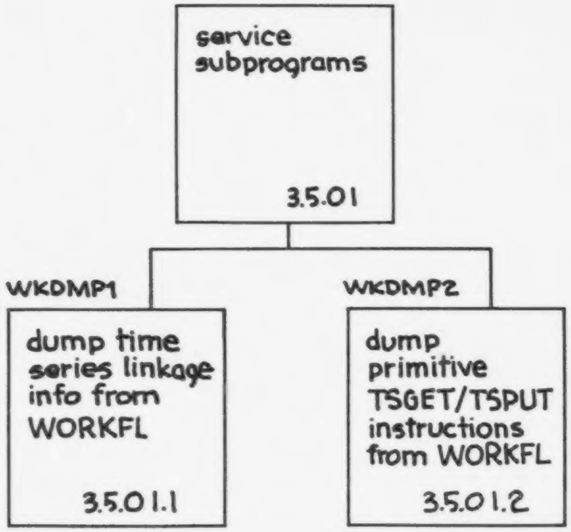
Structure chart 3.4(3). Processing of input for the RCHRES module


PHYDR
process input for Section HYDR
3.4(3).2
DISCH
compute initial discharge rates
3.4(3).2.1
FNDROW
4.2(3).1.2.1
AUXIL
4.2(3).1.3

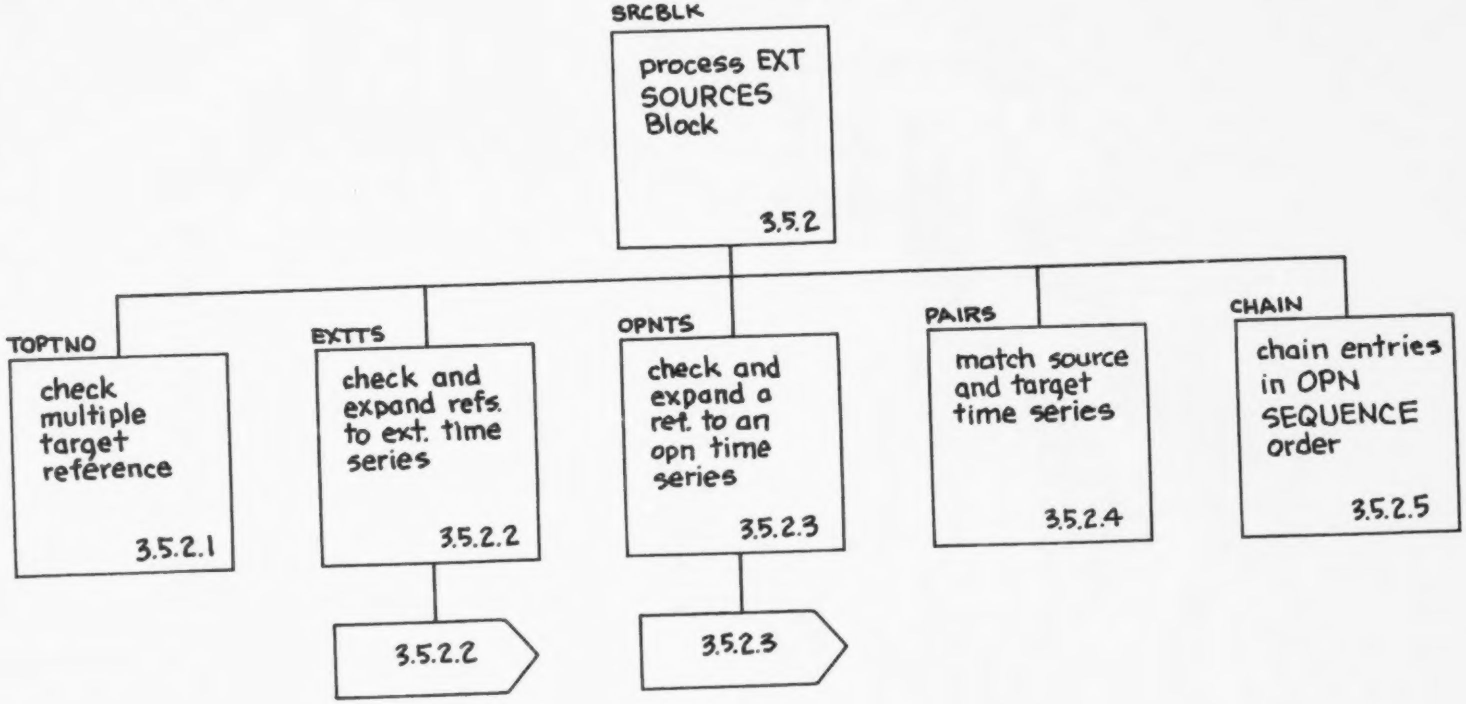
Structure chart 3.4(3).2 Processing of input for the HYDR section of the RCHRES module


TIMSER
process blocks which deal with time series
3.5
service subprograms
3.5.01
3.5.01
CHKTSS
look at the TSS descriptor
3.5.1.1
BPFINT
initialize TSGET/PUT instr. files
3.5.1.2
SRCBLK
process EXT SOURCES Block
3.5.2
3.5.2
NETBLK
process the NETWORK Block
3.5.3
3.5.3
TARBLK
process the EXT TARGETS Block
3.5.4
3.5.4
SAME
chain entries with same value
3.5.5
ALLOC
allocate rows in INPAD
3.5.6
3.5.6
OSVFPT
assign values to flag-ptrs. in OSVs
3.5.7
OSVPRO
process a group of entries in WORKFL
3.5.7.1
TINSTR
generate final TSGET/PUT instructions
3.5.8
3.5.8

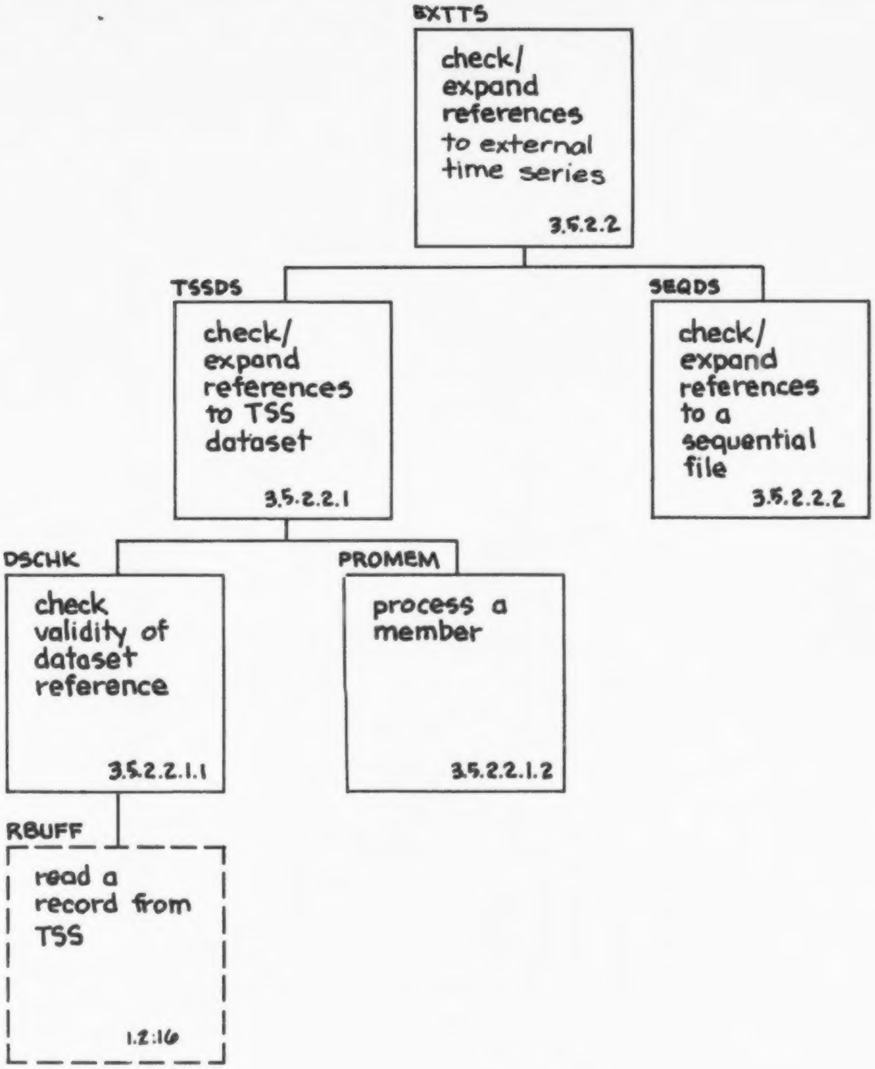
Structure chart 3.5 Processing of User's Control Input which deals with time series



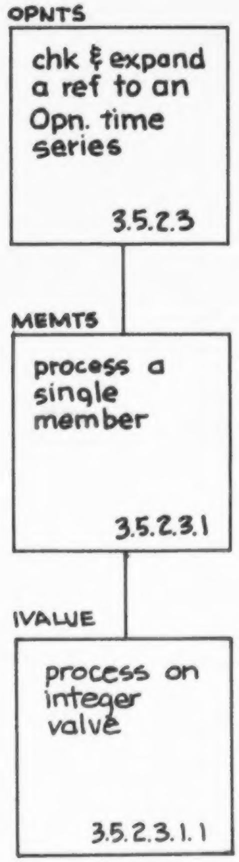
Structure chart 3.5.01 Service subprograms for the TIMSER subroutine group.


SRCBLK
process EXT SOURCES Block
3.5.2
TOPTNO
check multiple target reference
3.5.2.1
EXTTS
check and expand refs. to ext. time series
3.5.2.2
3.5.2.2
OPNTS
check and expand a ref. to an opn time series
3.5.2.3
3.5.2.3
PAIRS
match source and target time series
3.5.2.4
CHAIN
chain entries in OPN SEQUENCE order
3.5.2.5

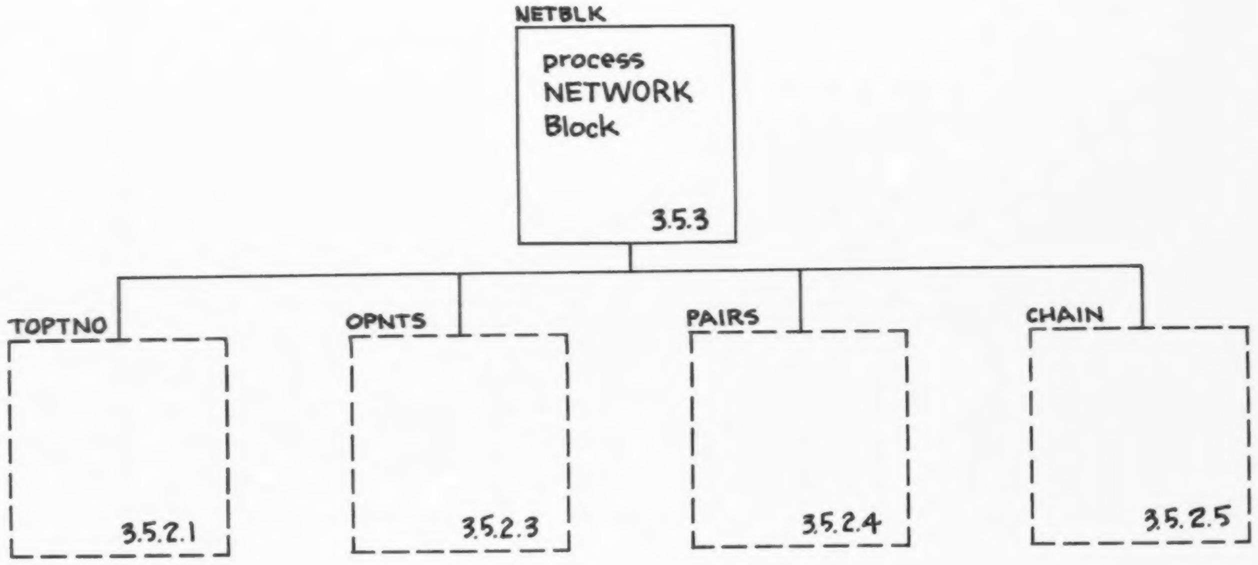
Structure chart 3.5.2 Processing entries in the EXT SOURCES Block


EXTTS
check/ expand references to external time series
3.5.2.2
TSSDS
check/ expand references to TSS dataset
3.5.2.2.1
SEQDS
check/ expand references to a sequential file
3.5.2.2.2
DSCHK
check validity of dataset reference
3.5.2.2.1.1
PROMEM
process a member
3.5.2.2.1.2
RBUFF
read a record from TSS
1.2:16

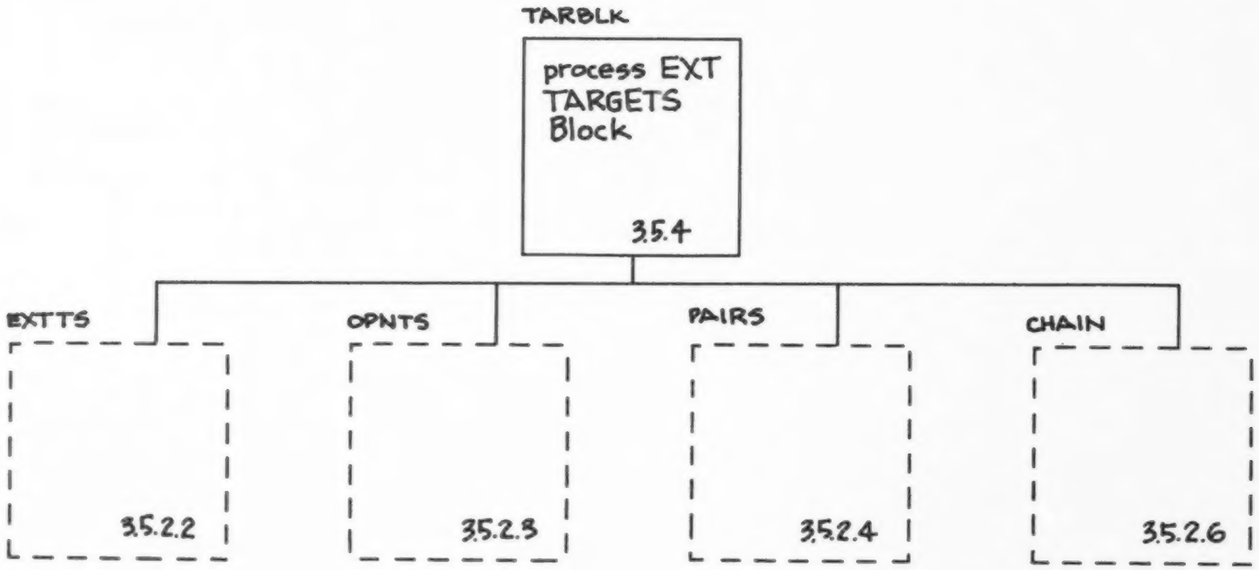
Structure chart 3.5.2.2 Subroutine group EXTTS



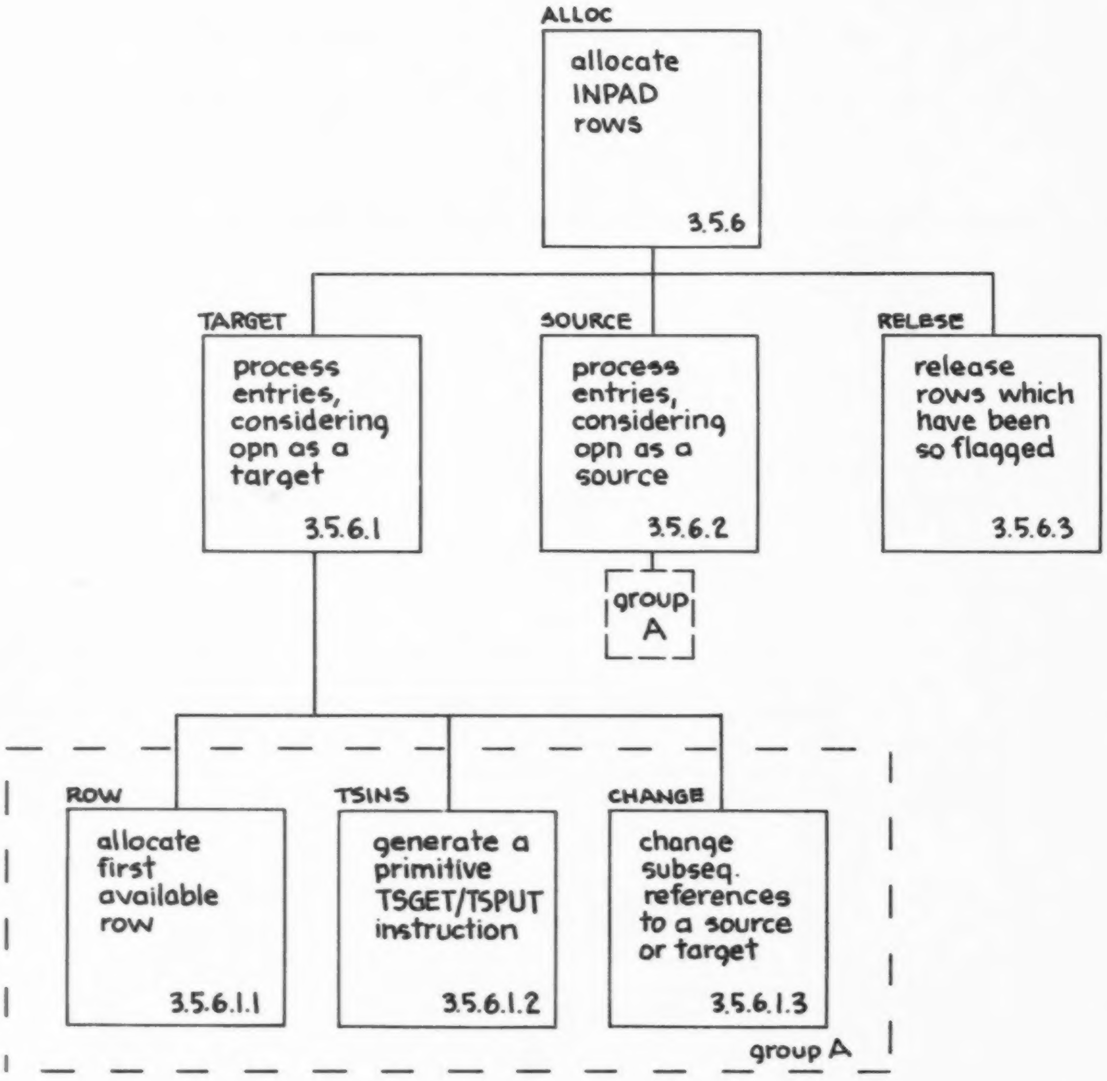
Structure chart 3.5.2.3 Check and expand a reference to an *Operation* time series



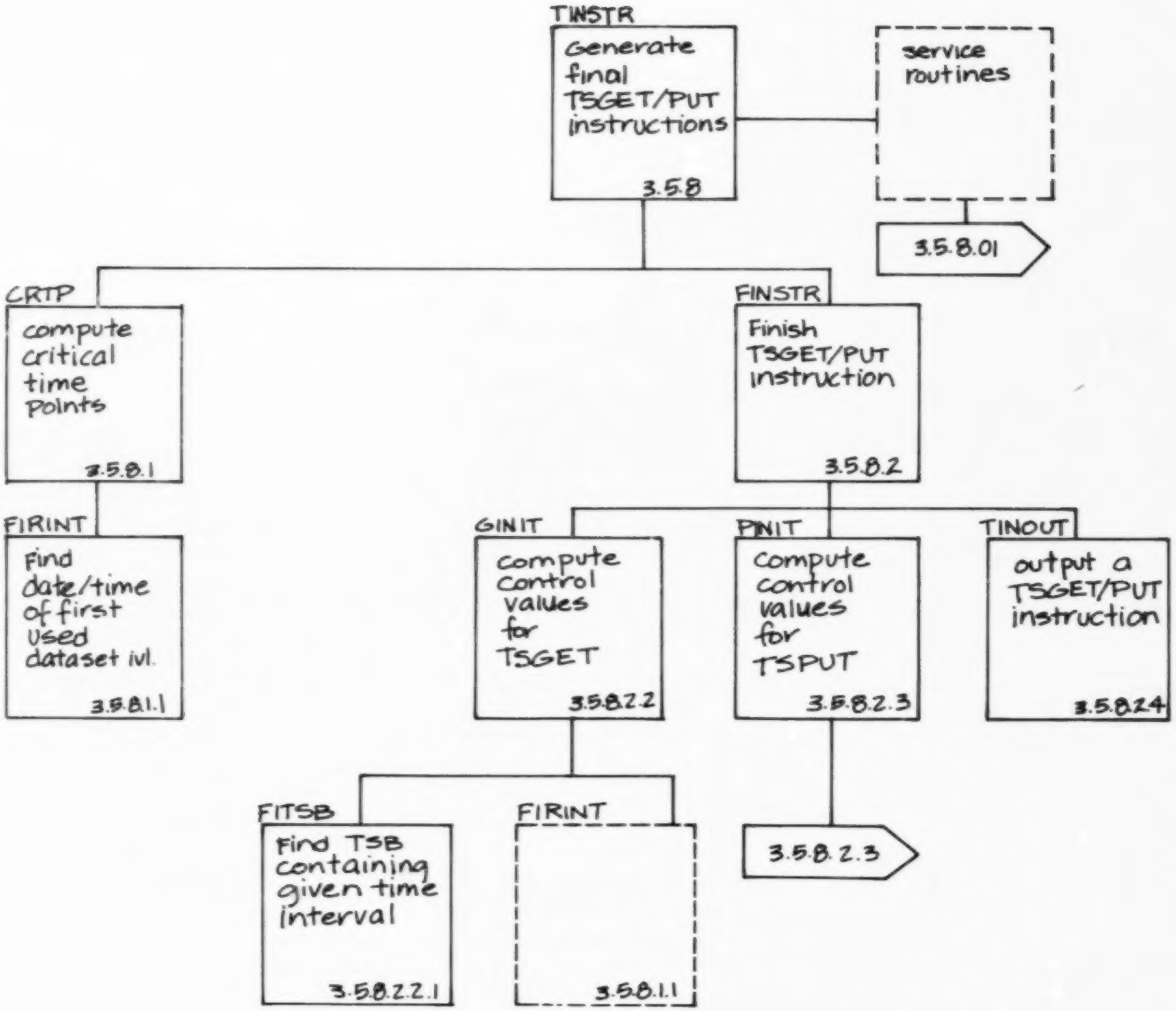
Structure chart 3.5.3 Processing entries in the NETWORK Block



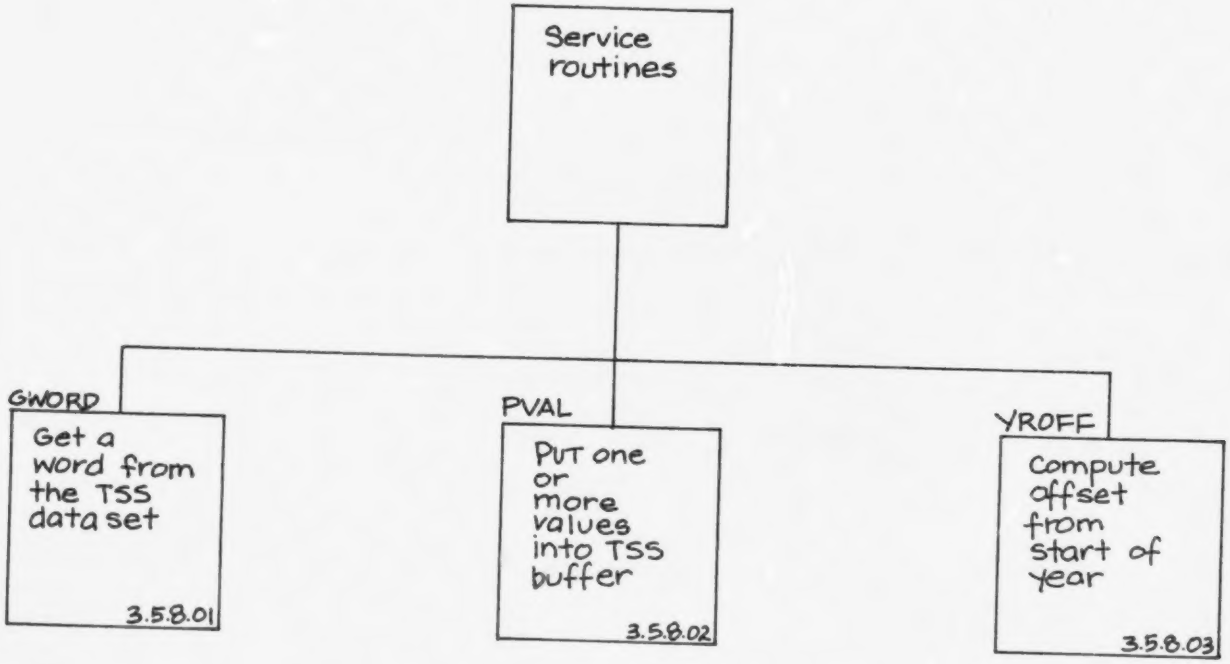
Structure chart 3.5.4 Processing entries in the EXT TARGETS Block


ALLOC
allocate INPAD rows
3.5.6
TARGET
process entries, considering opn as a target
3.5.6.1
SOURCE
process entries, considering opn as a source
3.5.6.2
group A
RELESE
release rows which have been so flagged
3.5.6.3
ROW
allocate first available row
3.5.6.1.1
TSINS
generate a primitive TSGET/TSPUT instruction
3.5.6.1.2
CHANGE
change subseq. references to a source or target
3.5.6.1.3
group A

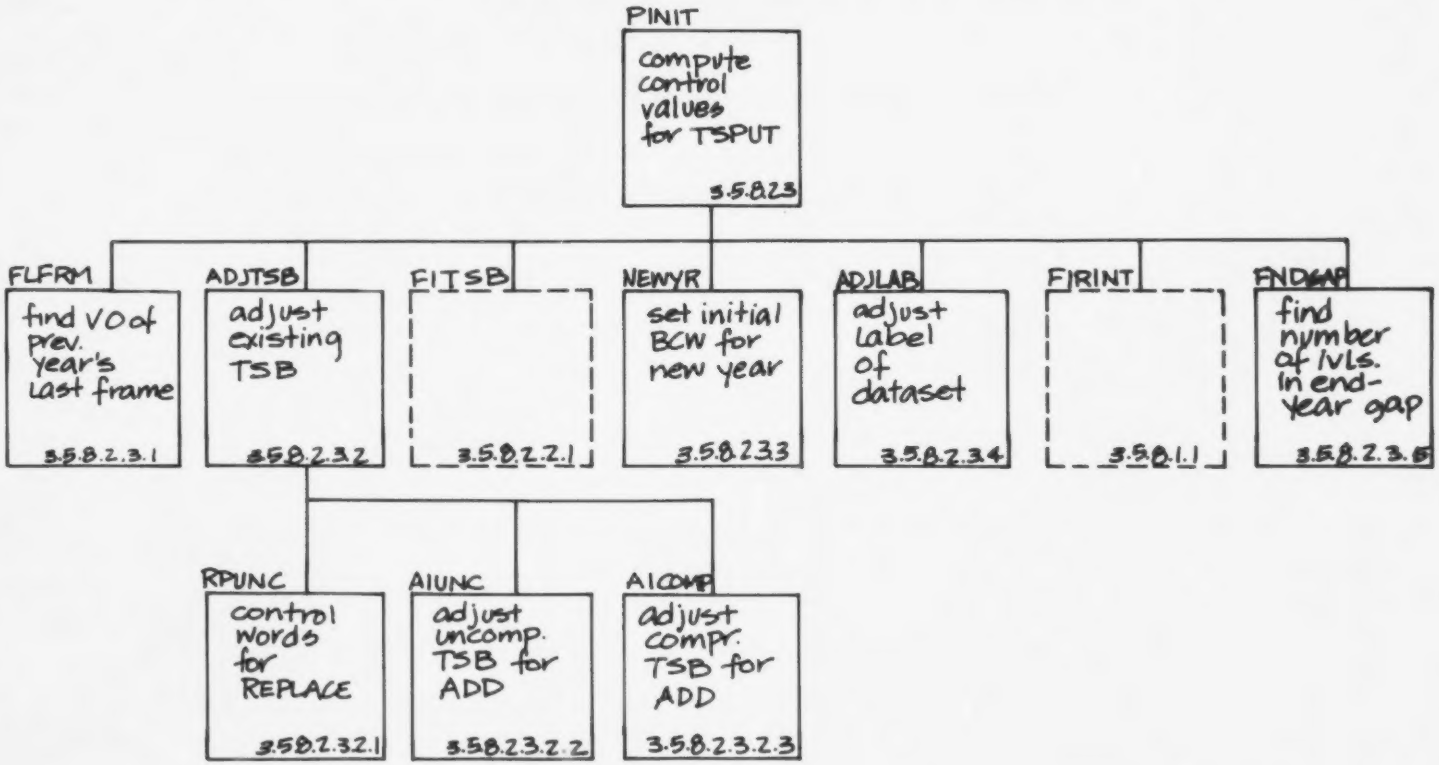
Structure chart 3.5.6 Subprograms involved in allocation & deallocation of INPAD rows


TINSTR
Generate final TSGET/PUT instructions
3.5.8
service routines
3.5.8.01
CRTP
compute critical time points
3.5.8.1
FINSTR
Finish TSGET/PUT instruction
3.5.8.2
FIRINT
Find date/time of first used dataset ivl.
3.5.8.1.1
GINIT
compute control values for TSGET
3.5.8.2.2
PINIT
compute control values for TSPUT
3.5.8.2.3
TINOUT
output a TSGET/PUT instruction
3.5.8.2.4
FITSB
Find TSB containing given time interval
3.5.8.2.2.1
FIRINT
3.5.8.1.1
3.5.8.2.3

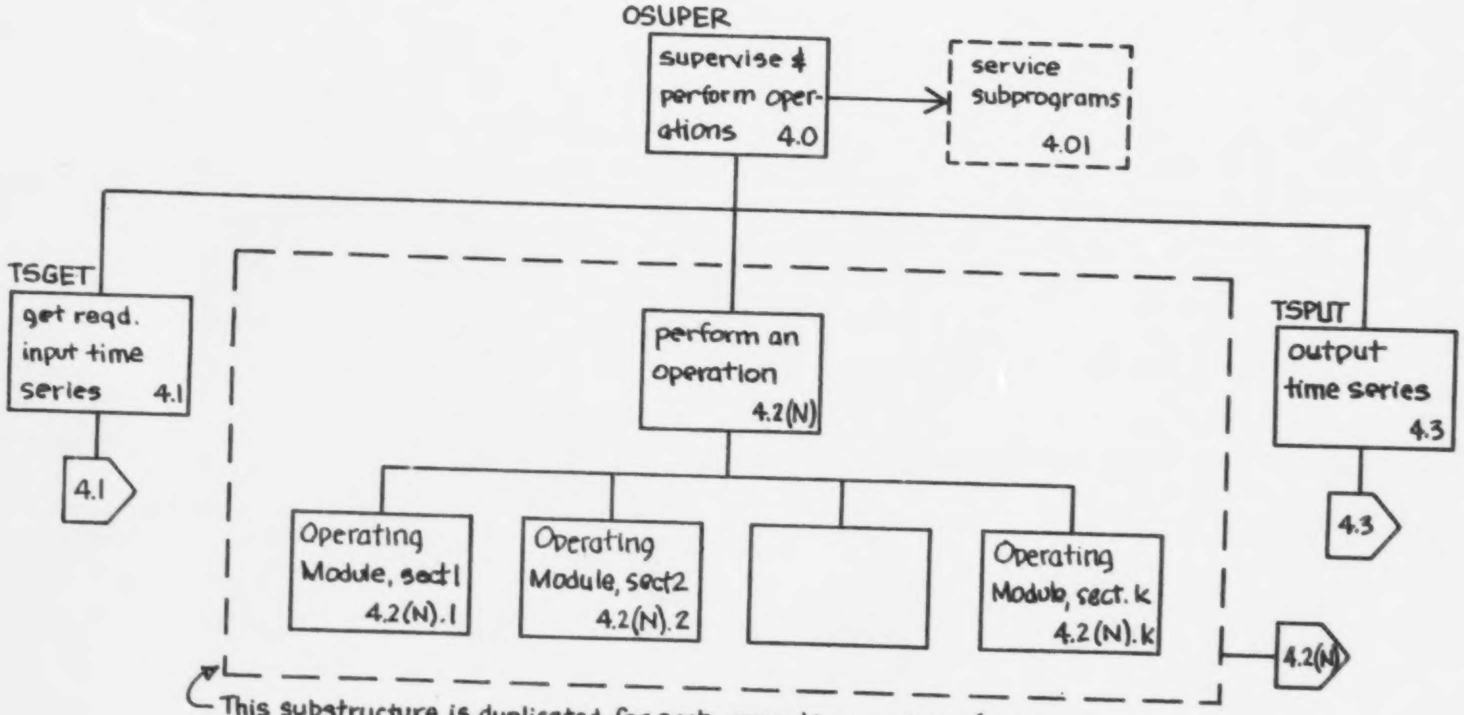
structure chart 3.5.8 Subroutine group TINSTR


Service routines
GWORD
Get a word from the TSS data set
3.5.8.01
PVAL
PUT one or more values into TSS buffer
3.5.8.02
YROFF
Compute offset from start of year
3.5.8.03

Structure Chart 3.5.8.01 Service routines for Subroutine group TINSTR



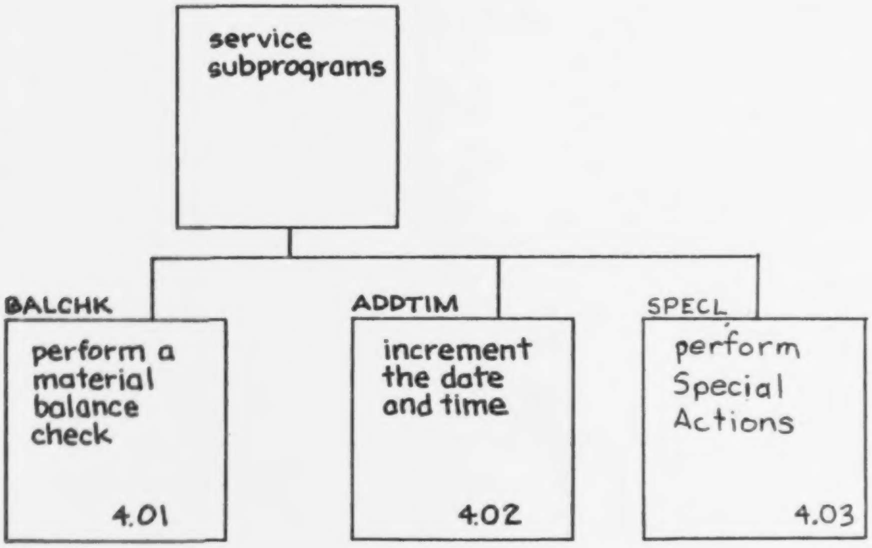
structure chart 3.5.8.2.3 subroutine group PINIT


OSUPER
supervise $ perform operations 4.0
service subprograms 4.01
TSGET
get reqd. input time series 4.1
4.1
perform an operation 4.2(N)
Operating Module, sect1 4.2(N).1
Operating Module, sect2 4.2(N).2
Operating Module, sect. k 4.2(N).k
4.2(N)
TSPUT
output time series 4.3
4.3

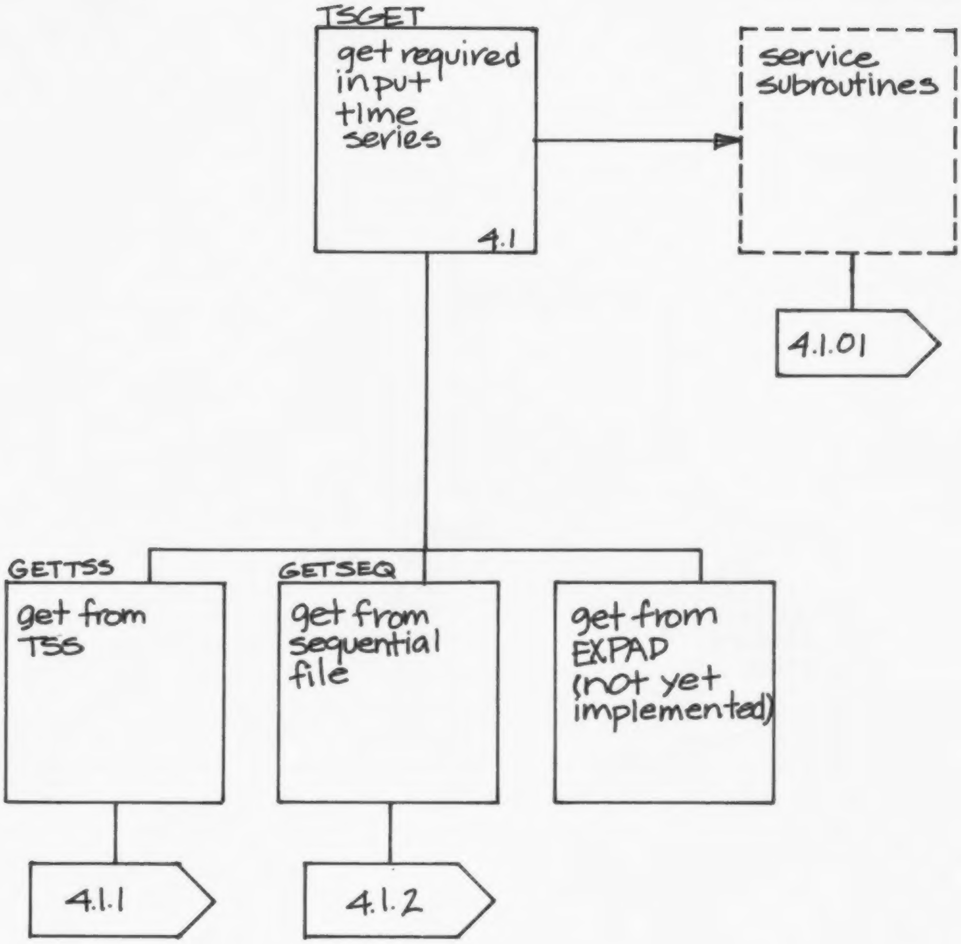
This substructure is duplicated for each operating module (utility or application module) in the system.

The numbering system shown is that for the Nth operating module

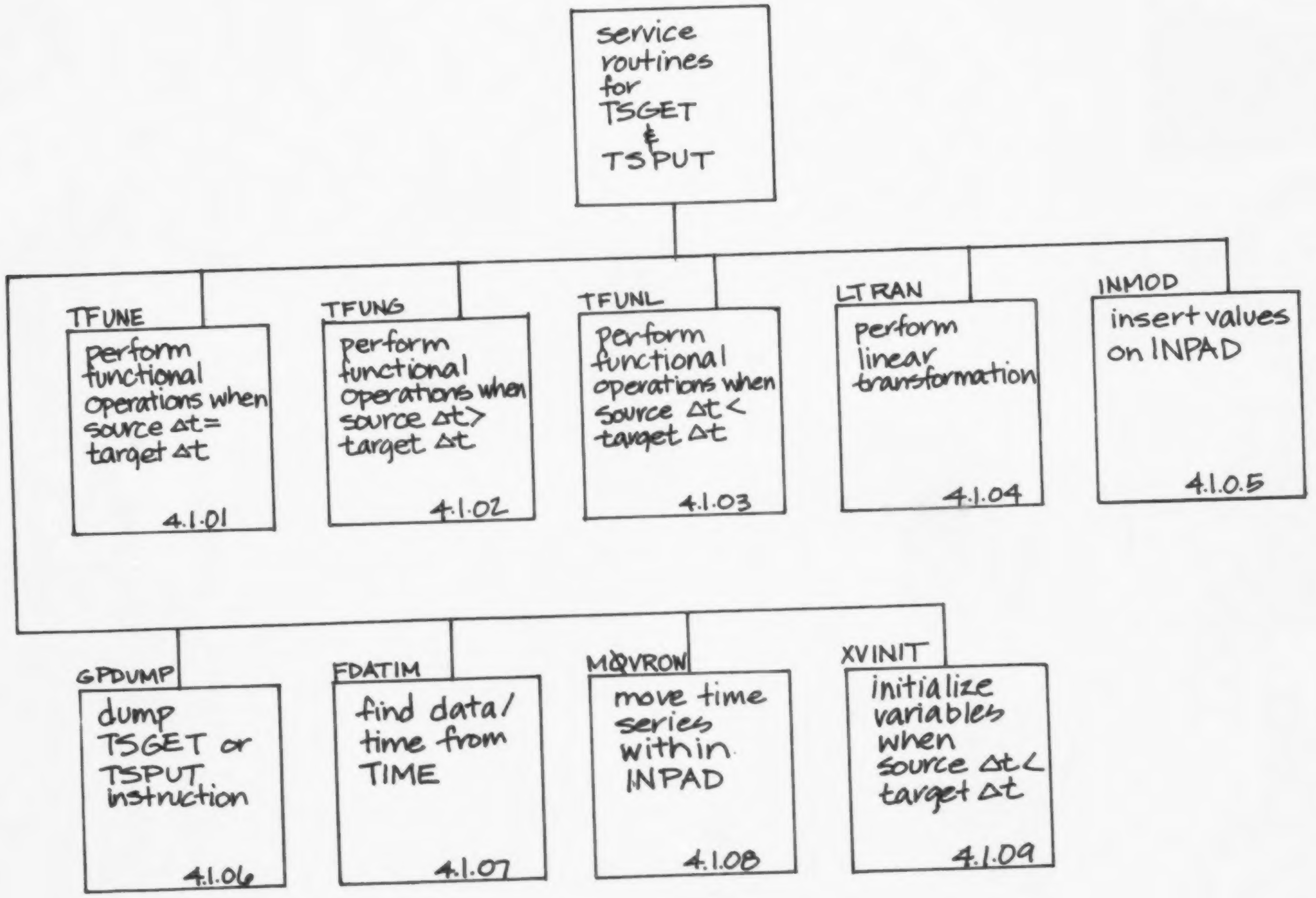
Structure chart 4.0 Operations group of modules



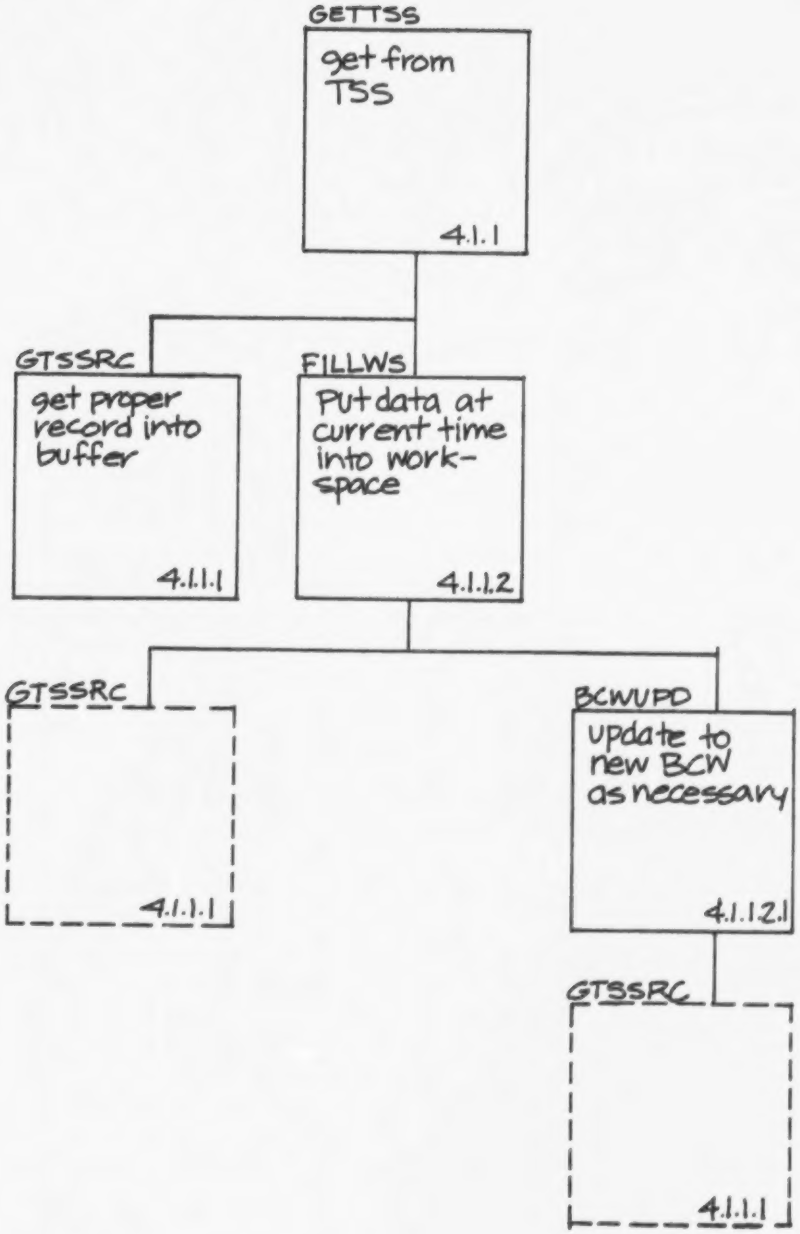
Structure chart 4.01 Service subprograms for the Operations Supervisor



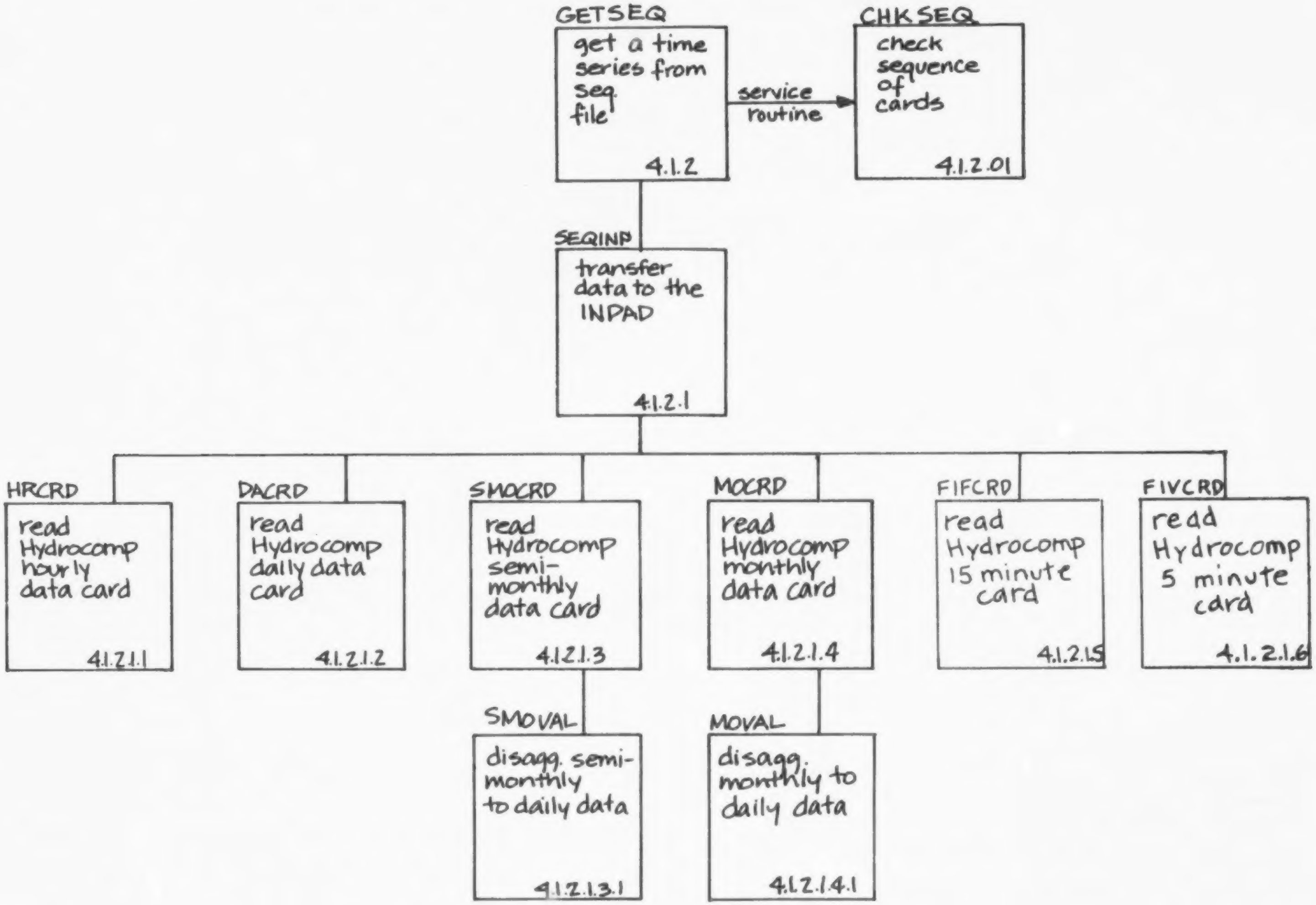
structure chart 4.1 TSGET


service routines for TSGET & TSPUT
TFUNE
perform functional operations when source Δt= target Δt
4.1.01
TFUNG
perform functional operations when source Δt> target Δt
4.1.02
TFUNL
perform functional operations when source Δt< target Δt
4.1.03
LTRAN
perform linear transformation
4.1.04
INMOD
insert values on INPAD
4.1.0.5
GPDUMP
dump TSGET or TSPUT instruction
4.1.06
FDATIM
find data/ time from TIME
4.1.07
MOVROW
move time series within INPAD
4.1.08
XVINIT
initialize variables when source Δt< target Δt
4.1.09

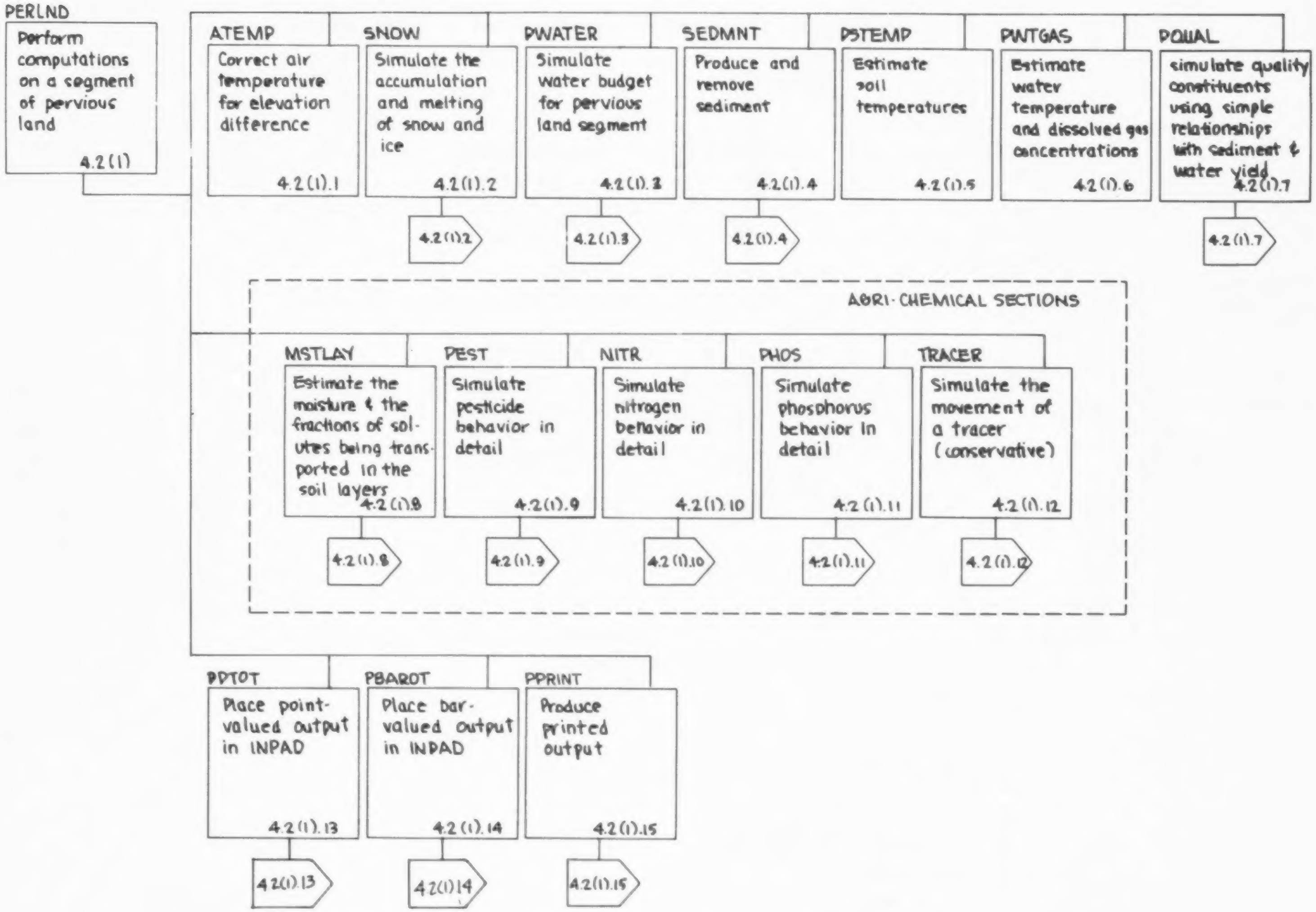
structure chart 4.1.01 service routines for TSGET and TSPUT


GETTSS
get from TSS
4.1.1
GTSSRC
get proper record into buffer
4.1.1.1
FILLWS
Put data at current time into work-space
4.1.1.2
GTSSRC
4.1.1.1
BCWUPD
update to new BCW as necessary
4.1.1.2.1
GTSSRC
4.1.1.1

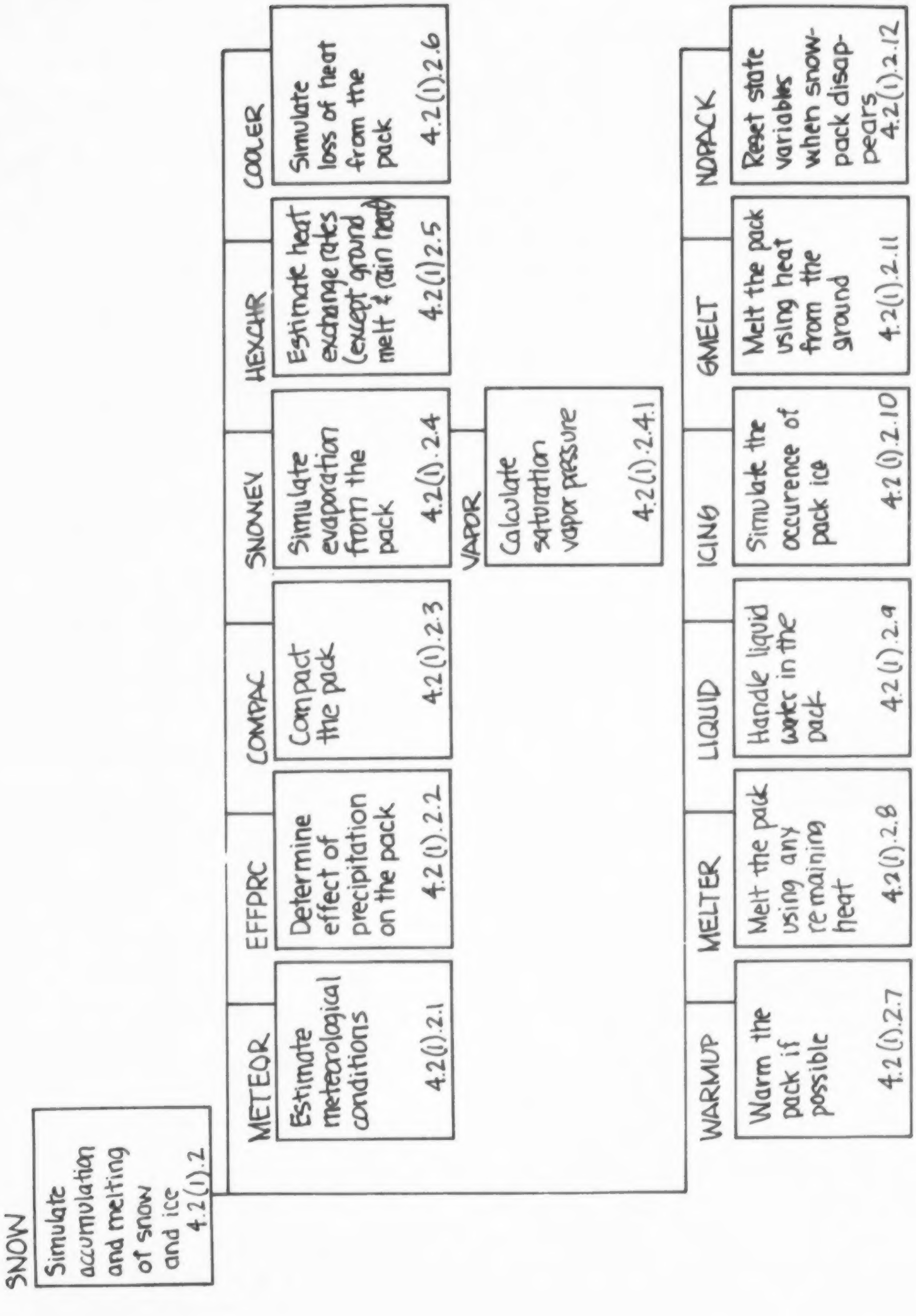
structure chart 4.1.1 The GETTSS section of the TSGET module


GETSEQ
get a time series from seq. file
4.1.2
service routine
CHKSEQ
check sequence of cards
4.1.2.01
SEQINP
transfer data to the INPAD
4.1.2.1
HRCRD
read Hydrocomp hourly data card
4.1.2.1.1
DACRD
read Hydrocomp daily data card
4.1.2.1.2
SMOCRD
read Hydrocomp semi-monthly data card
4.1.2.1.3
MOCRD
read Hydrocomp monthly data card
4.1.2.1.4
FIFCRD
read Hydrocomp 15 minute card
4.1.2.1.5
FIVCRD
read Hydrocomp 5 minute card
4.1.2.1.6
SMOVAL
disagg. semi-monthly to daily data
4.1.2.1.3.1
MOVAL
disagg. monthly to daily data
4.1.2.1.4.1

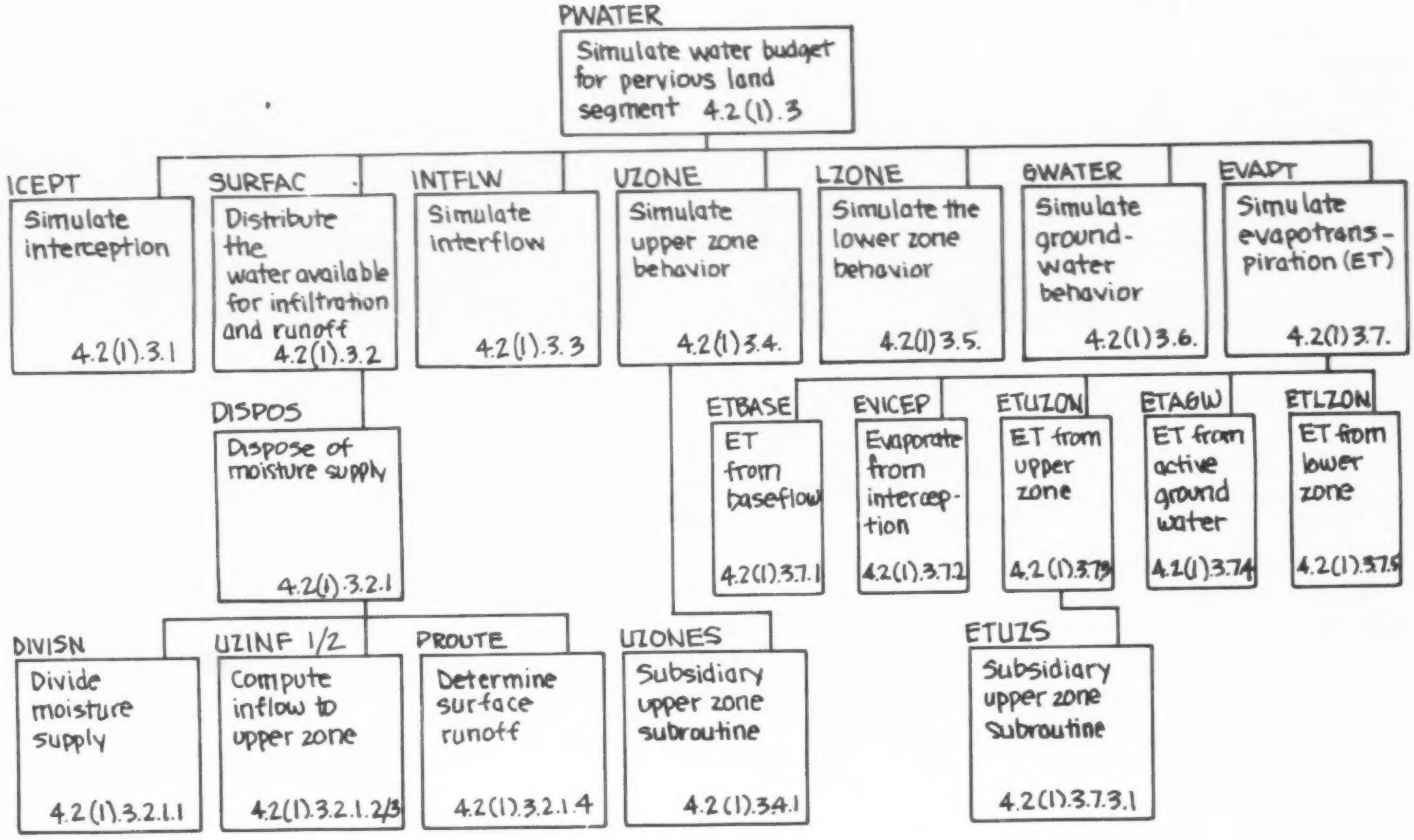
Structure chart 4.1.2 The GETSEQ section of the TSGET module.


PERLND
Perform computations on a segment of pervious land
4.2(1)
ATEMP
Correct air temperature for elevation difference
4.2(1).1
SNOW
Simulate the accumulation and melting of snow and ice
4.2(1).2
4.2(1).2
PWATER
Simulate water budget for pervious land segment
4.2(1).3
4.2(1).3
SEDMNT
Produce and remove sediment
4.2(1).4
4.2(1).4
PSTEMP
Estimate soil temperatures
4.2(1).5
PWTGAS
Estimate water temperature and dissolved gas concentrations
4.2(1).6
PQUAL
simulate quality constituents using simple relationships with sediment & water yield
4.2(1).7
4.2(1).7
AGRI-CHEMICAL SECTIONS
MSTLAY
Estimate the moisture & the fractions of solutes being transported in the soil layers
4.2(1).8
4.2(1).8
PEST
Simulate pesticide behavior in detail
4.2(1).9
4.2(1).9
NITR
Simulate nitrogen behavior in detail
4.2(1).10
4.2(1).10
PHOS
Simulate phosphorus behavior in detail
4.2(1).11
4.2(1).11
TRACER
Simulate the movement of a tracer (conservative)
4.2(1).12
4.2(1).12
PPTOT
Place point-valued output in INPAD
4.2(1).13
4.2(1).13
PBAROT
Place bar-valued output in INPAD
4.2(1).14
4.2(1).14
PPRINT
Produce printed output
4.2(1).15
4.2(1).15

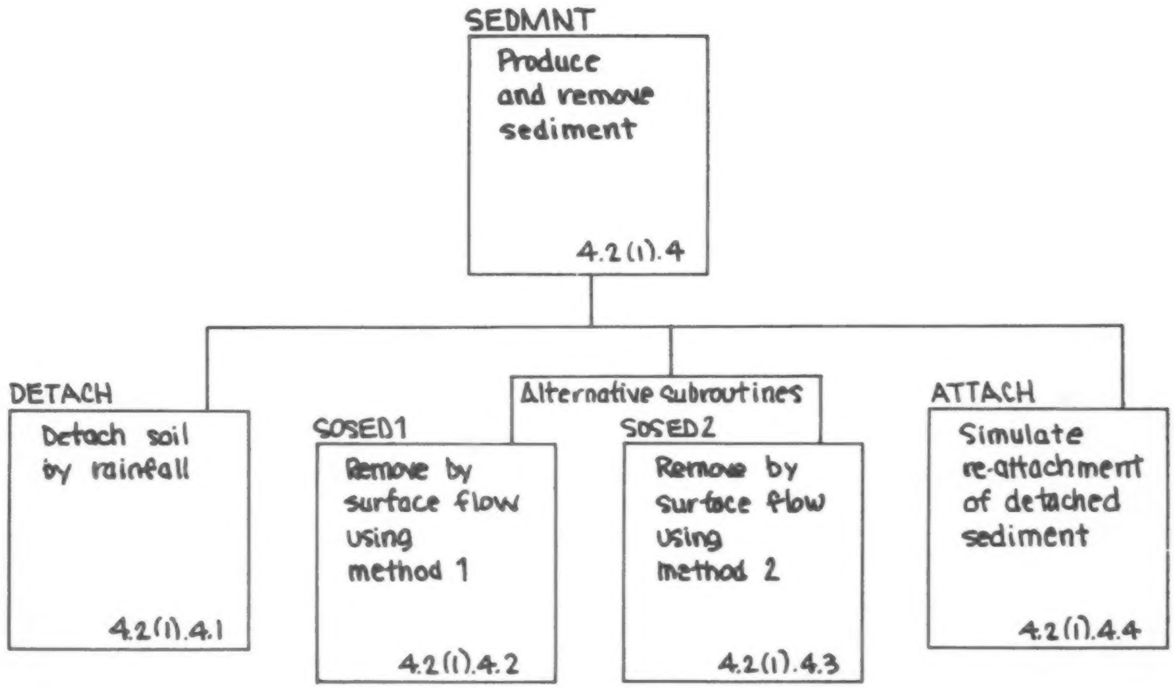
Structure chart 4.2(1) Pervious land-segment application module


SNOW
Simulate accumulation and melting of snow and ice
4.2(1).2
METEOR
Estimate meteorological conditions
4.2(1).2.1
EFFPRC
Determine effect of precipitation on the pack
4.2(1).2.2
COMPAC
Compact the pack
4.2(1).2.3
SNOWEV
Simulate evaporation from the pack
4.2(1).2.4
HEXCHR
Estimate heat exchange rates (except ground melt & rain heat)
4.2(1).2.5
COOLER
Simulate loss of heat from the pack
4.2(1).2.6
VAPOR
Calculate saturation vapor pressure
4.2(1).2.4.1
WARMUP
Warm the pack if possible
4.2(1).2.7
MELTER
Melt the pack using any remaining heat
4.2(1).2.8
LIQUID
Handle liquid water in the pack
4.2(1).2.9
ICING
Simulate the occurence of pack ice
4.2(1).2.10
GMELT
Melt the pack using heat from the ground
4.2(1).2.11
NOPACK
Reset state variables when snowpack disappears
4.2(1).2.12

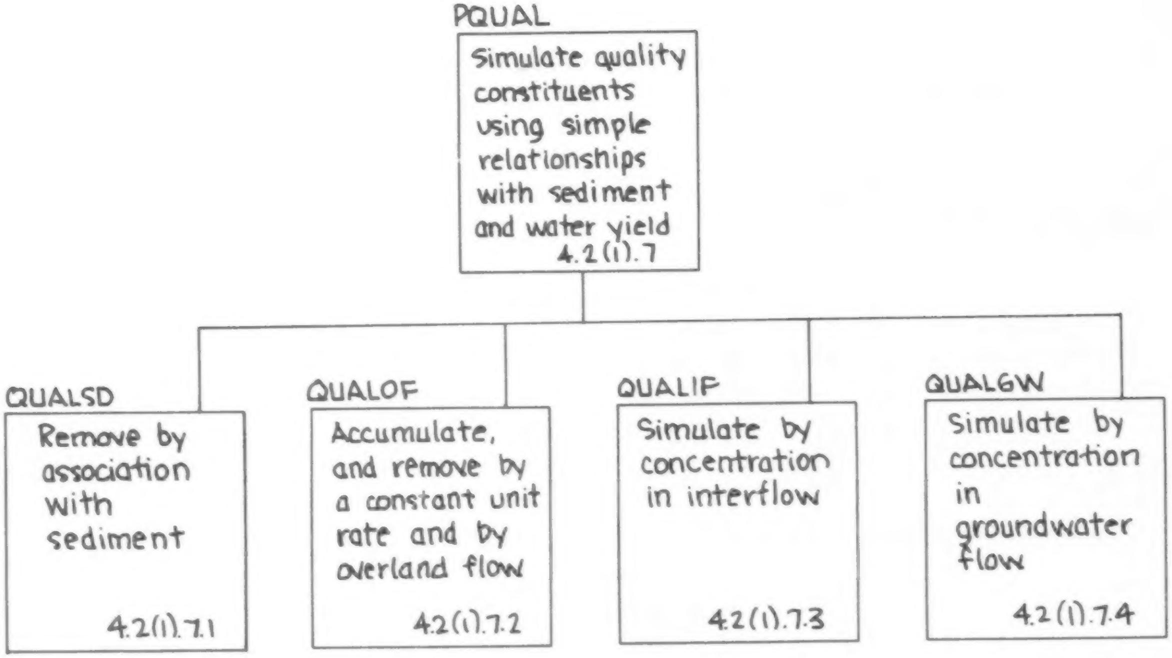
Structure chart 4.2(1).2 The SNOW section of modules PERLND and IMPLND



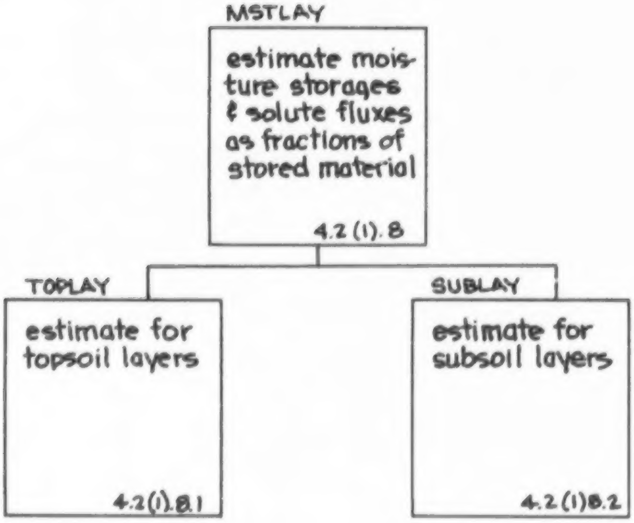
Structure Chart 4.2(1).3 The PWATER section of the PERLND application module


SEDMNT
Produce and remove sediment
4.2(1).4
DETACH
Detach soil by rainfall
4.2(1).4.1
Alternative subroutines
SOSED1
Remove by surface flow using method 1
4.2(1).4.2
SOSED2
Remove by surface flow using method 2
4.2(1).4.3
ATTACH
Simulate re-attachment of detached sediment
4.2(1).4.4

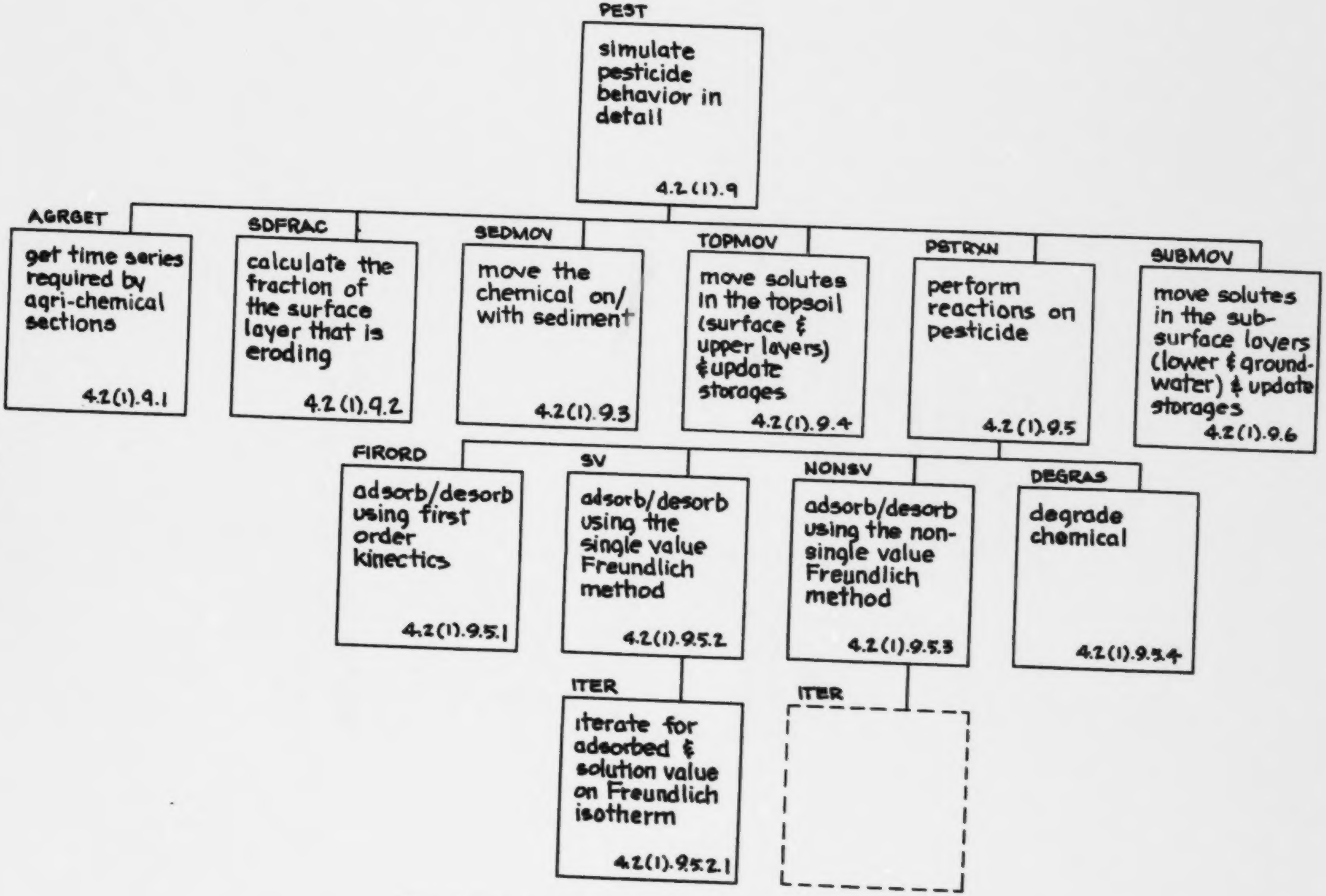
Structure chart 4.2(1).4 The SEDMNT section of the PERLND application module


PQUAL
Simulate quality constituents using simple relationships with sediment and water yield
4.2(1).7
QUALSD
Remove by association with sediment
4.2(1).7.1
QUALOF
Accumulate, and remove by a constant unit rate and by overland flow
4.2(1).7.2
QUALIF
Simulate by concentration in interflow
4.2(1).7.3
QUALGW
Simulate by concentration in groundwater flow
4.2(1).7.4

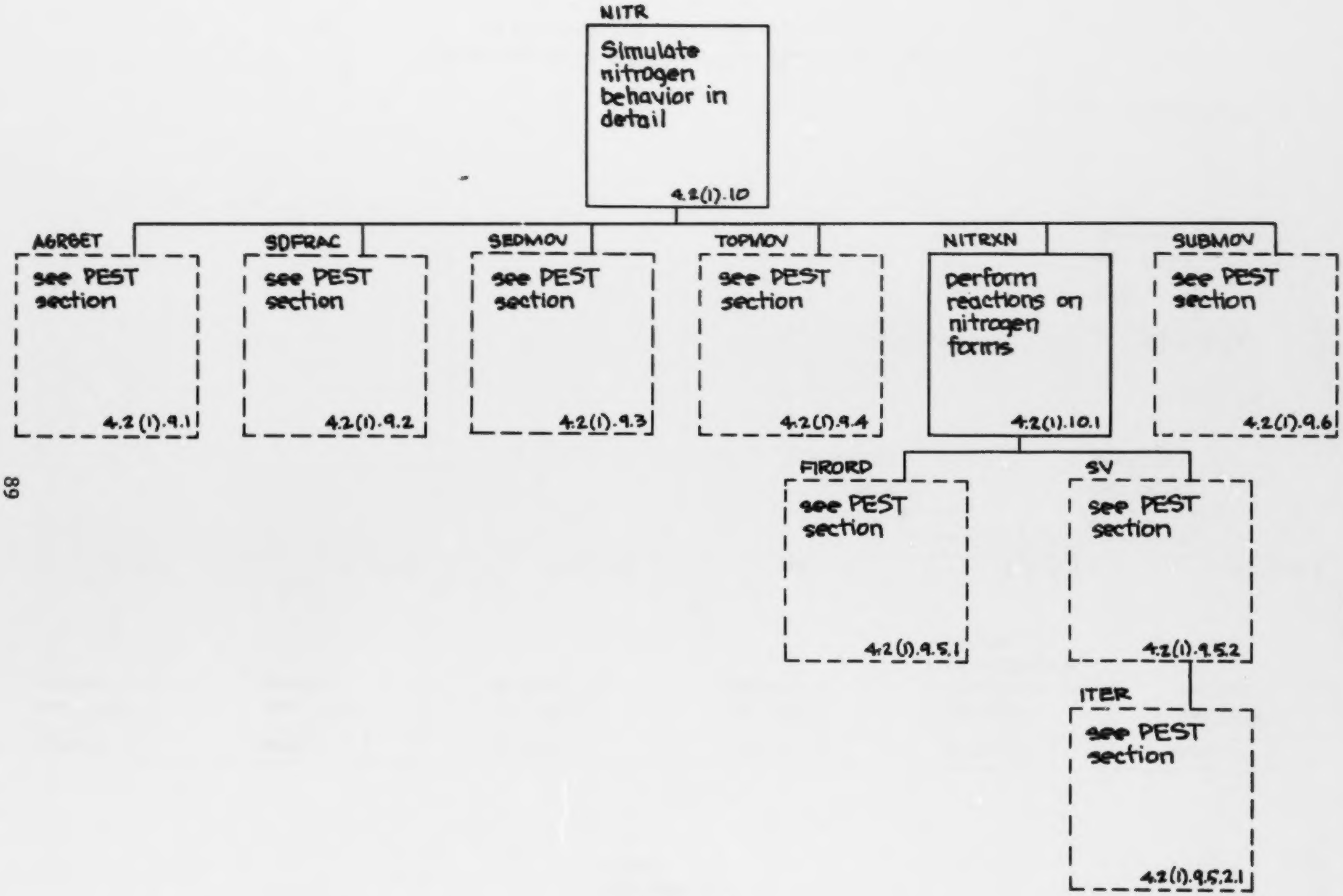
Structure chart 4.2(1).7 The PQUAL section of the PERLND application module


MSTLAY
estimate moisture storages & solute fluxes as fractions of stored material
4.2(1).8
TOPLAY
estimate for topsoil layers
4.2(1).8.1
SUBLAY
estimate for subsoil layers
4.2(1)8.2

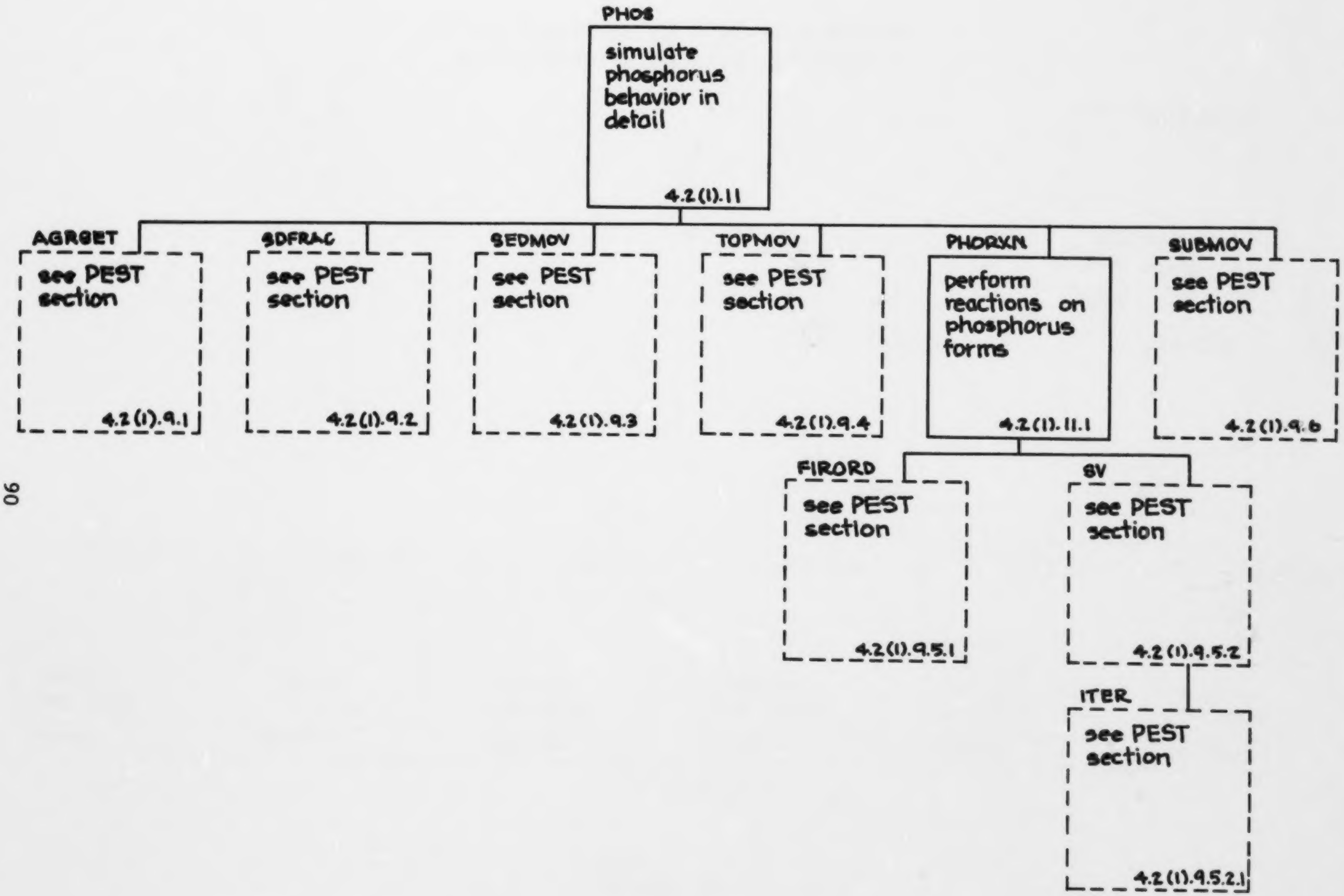
Structure chart 4.2(1).8 The MSTLAY section of the PERLND application module


PEST
simulate pesticide behavior in detail
4.2(1).9
AGRGET
get time series required by agri-chemical sections
4.2(1).9.1
SDFRAC
calculate the fraction of the surface layer that is eroding
4.2(1).9.2
SEDMOV
move the chemical on/with sediment
4.2(1).9.3
TOPMOV
move solutes in the topsoil (surface & upper layers) & update storages
4.2(1).9.4
PSTRXN
perform reactions on pesticide
4.2(1).9.5
SUBMOV
move solutes in the sub-surface layers (lower & ground-water) & update storages
4.2(1).9.6
FIRORD
adsorb/desorb using first order kinectics
4.2(1).9.5.1
SV
adsorb/desorb using the single value Freundlich method
4.2(1).9.5.2
NONSV
adsorb/desorb using the non-single value Freundlich method
4.2(1).9.5.3
DEGRAS
degrade chemical
4.2(1).9.5.4
ITER
iterate for adsorbed & solution value on Freundlich isotherm
4.2(1).9.5.2.1
ITER

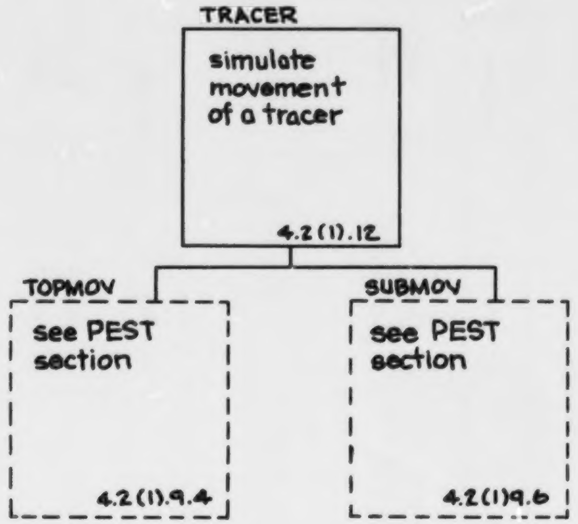
Structure chart 4.2(1).9 The PEST section of the PERLND application module



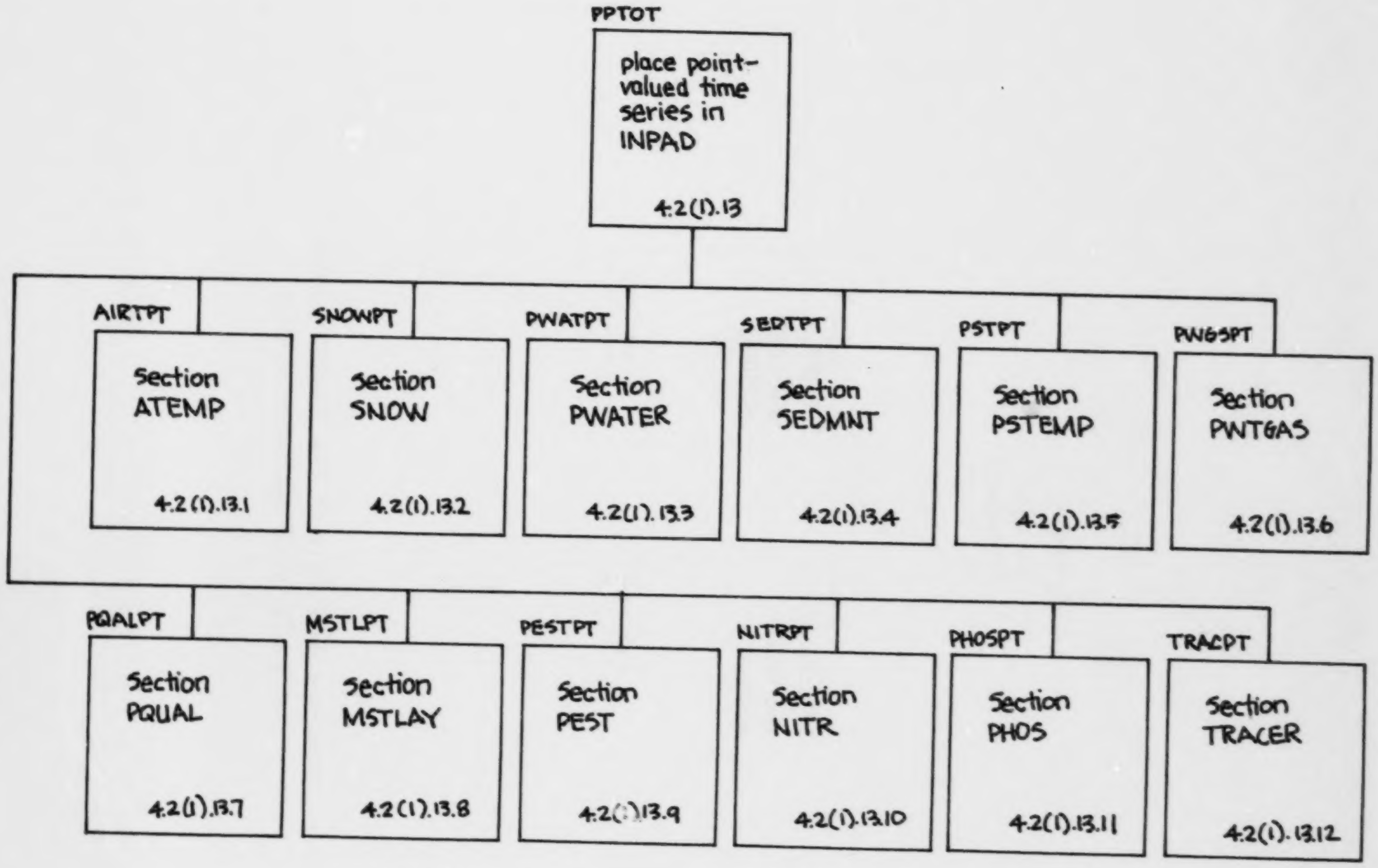
Structure chart 4.2(1).10 The NITR section of the PERLND application module



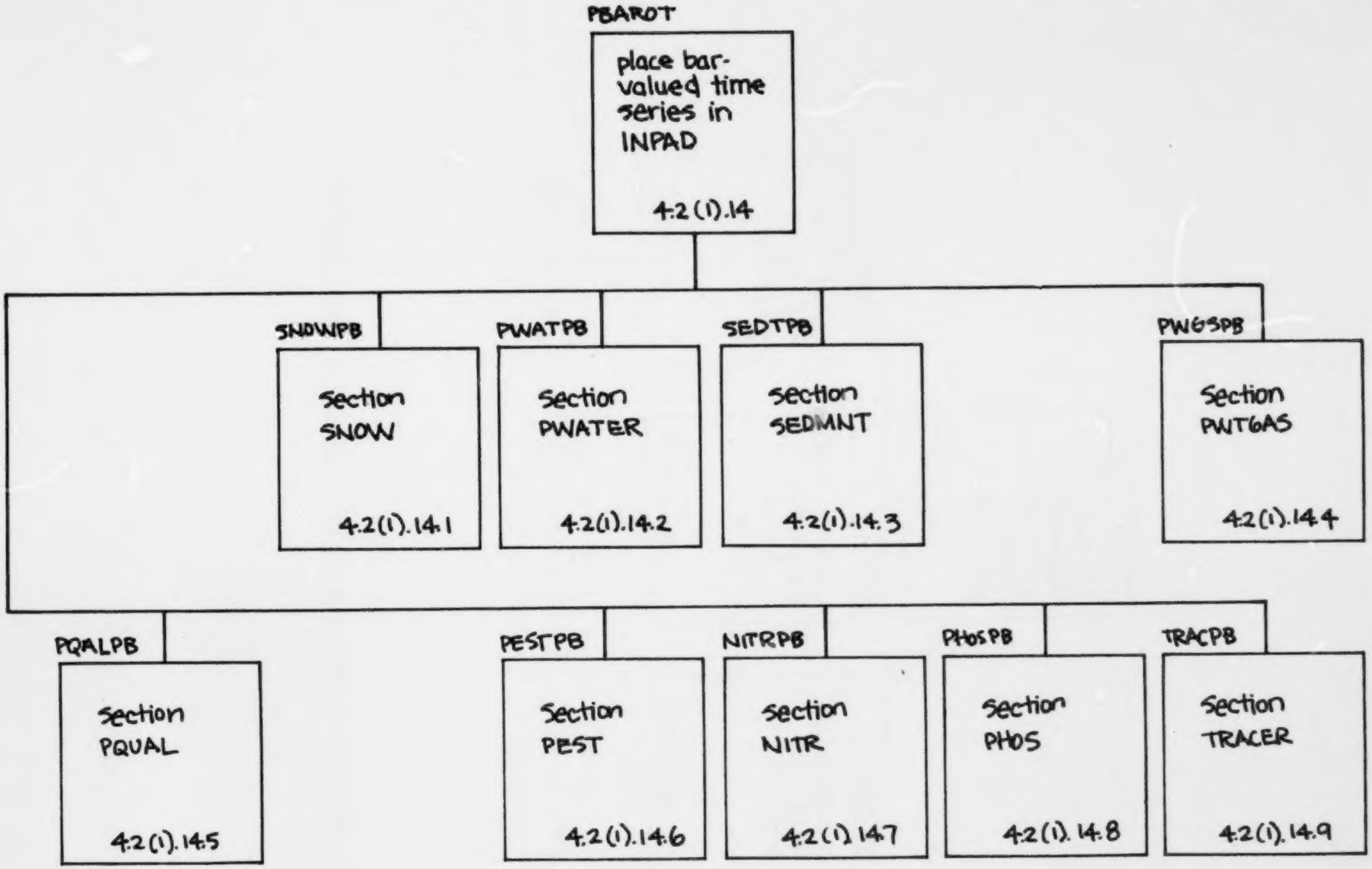
Structure chart 4.2 (1).11 The PHOS section of the PERLND application module



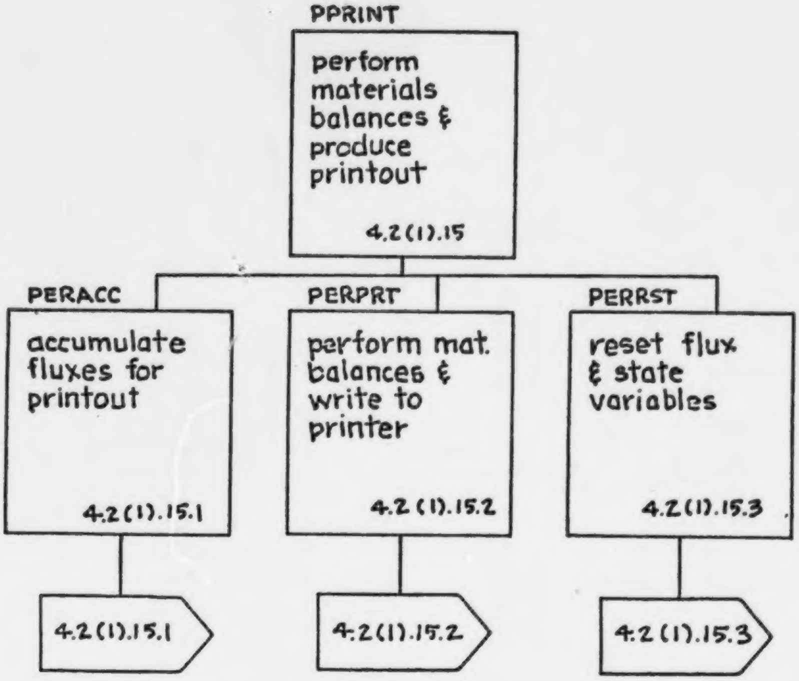
Structure chart 4.2(1).12 The TRACER section of the PERLND application module



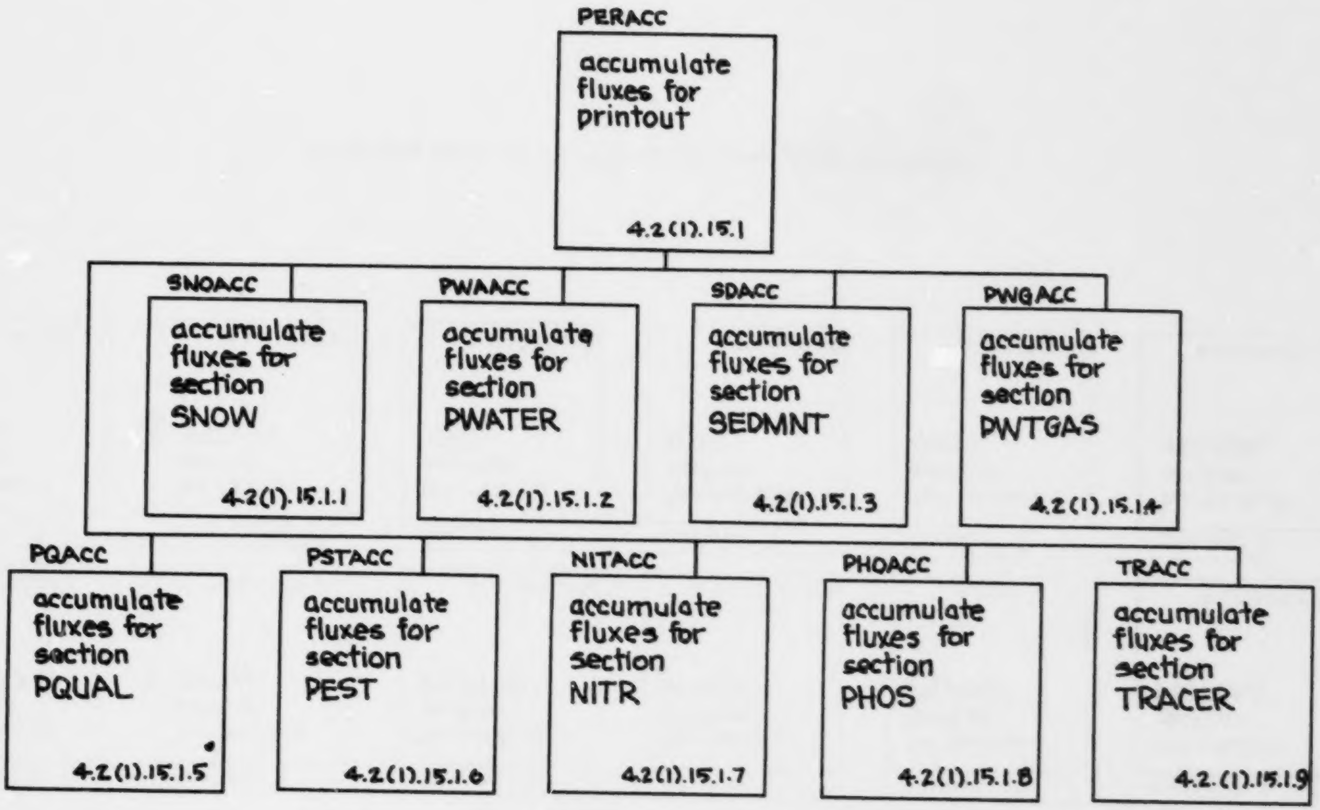
Structure chart 4.2(1).13 Subroutine group PPTOT



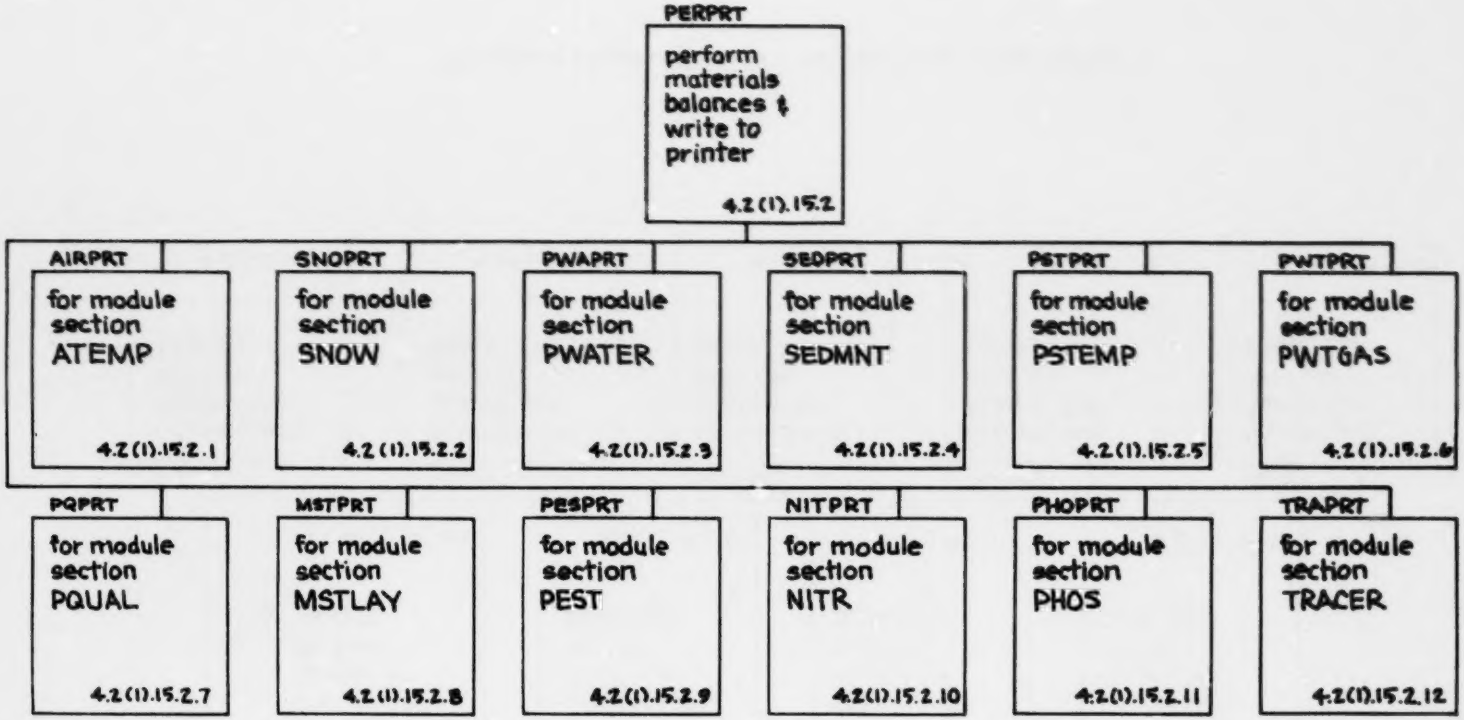
Structure chart 4:2(1).14 Subroutine group PBAROT


PPRINT
perform materials balances & produce printout
4.2(1).15
PERACC
accumulate fluxes for printout
4.2(1).15.1
PERPRT
perform mat. balances & write to printer
4.2(1).15.2
PERRST
reset flux & state variables
4.2(1).15.3
4.2(1).15.1
4.2(1).15.2
4.2(1).15.3

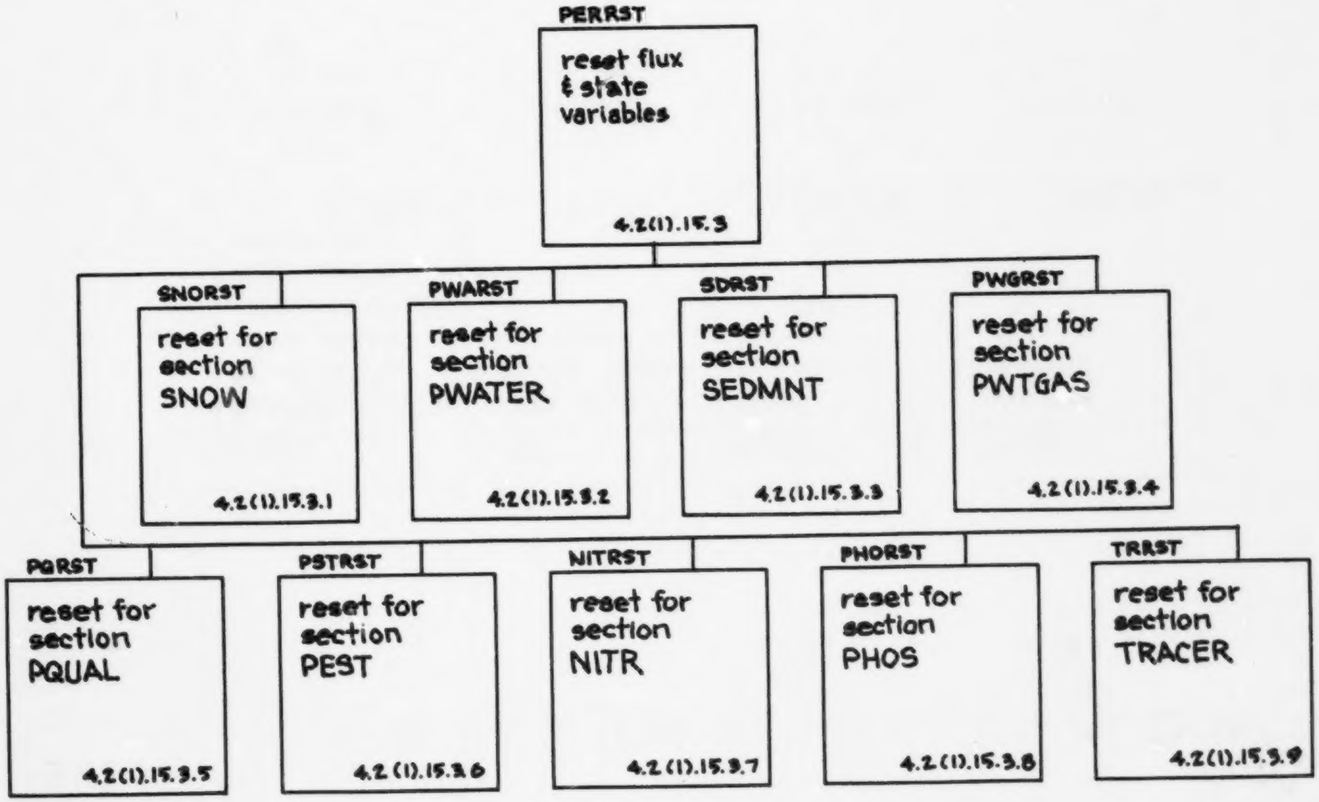
Structure chart 4.2(1).15 Subroutine group PPRINT


PERACC
accumulate fluxes for printout
4.2(1).15.1
SNOACC
accumulate fluxes for section SNOW
4.2(1).15.1.1
PWAACC
accumulate fluxes for section PWATER
4.2(1).15.1.2
SDACC
accumulate fluxes for section SEDMNT
4.2(1).15.1.3
PWGACC
accumulate fluxes for section PWTGAS
4.2(1).15.1.4
PQACC
accumulate fluxes for section PQUAL
4.2(1).15.1.5
PSTACC
accumulate fluxes for section PEST
4.2(1).15.1.6
NITACC
accumulate fluxes for section NITR
4.2(1).15.1.7
PHOACC
accumulate fluxes for section PHOS
4.2(1).15.1.8
TRACC
accumulate fluxes for section TRACER
4.2(1).15.1.9

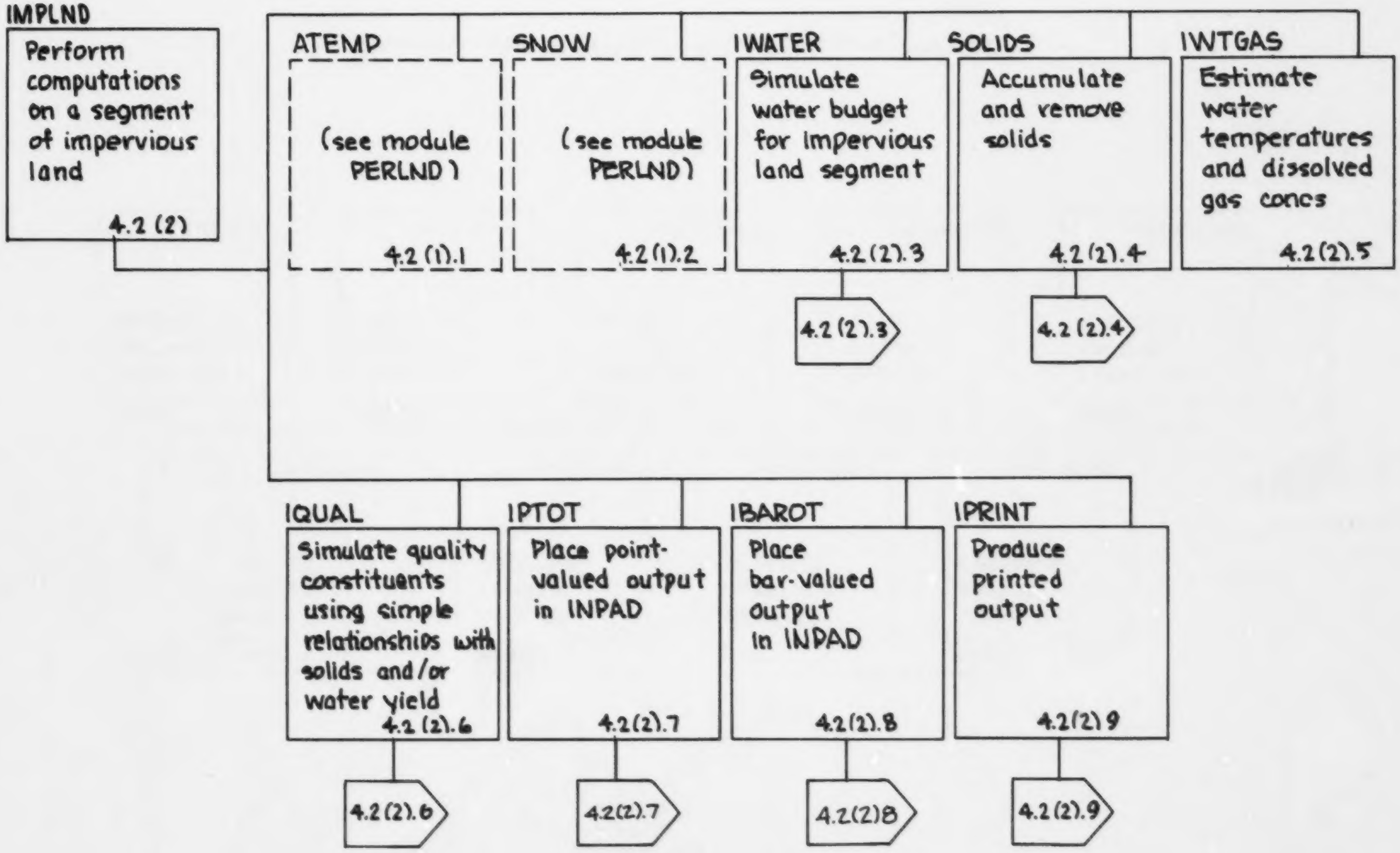
Structure chart 4.2(1).15.1 Subroutine group PERACC



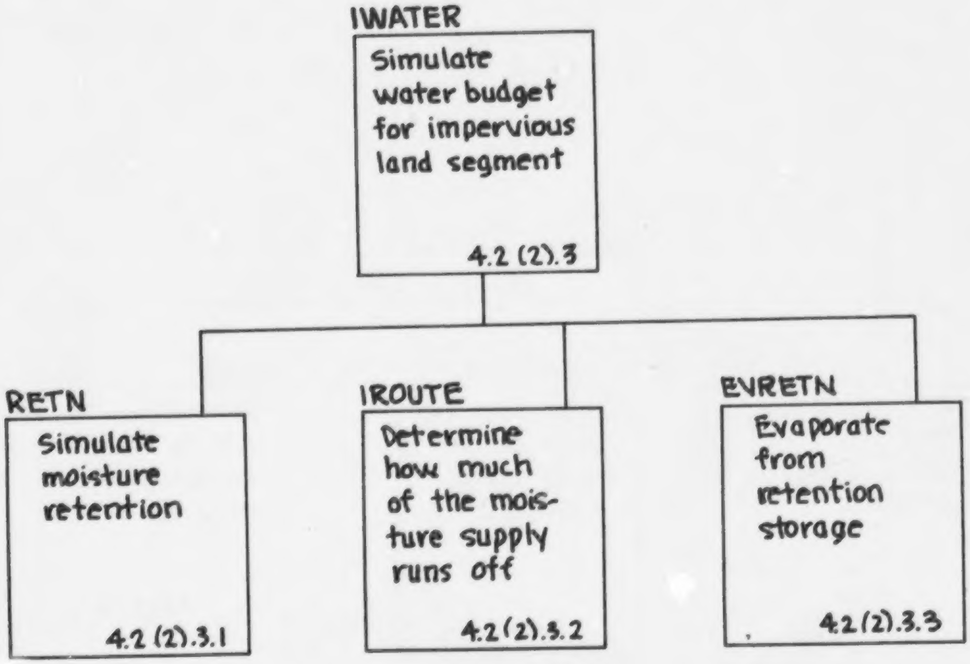
Structure chart 4.2 (1).15.2 Subroutine group PERPRT



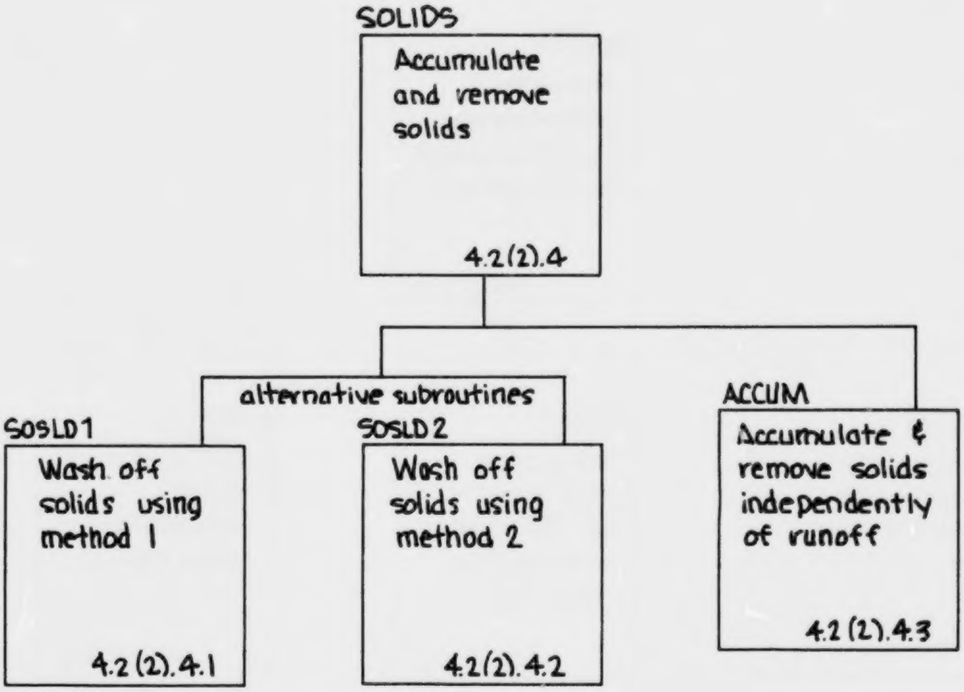
Structure chart 4.2(1).15.3 Subroutine group PERRST


IMPLND
Perform computations on a segment of impervious land
4.2 (2)
ATEMP
(see module PERLND)
4.2 (1).1
SNOW
(see module PERLND)
4.2 (1).2
IWATER
Simulate water budget for impervious land segment
4.2 (2).3
4.2 (2).3
SOLIDS
Accumulate and remove solids
4.2 (2).4
4.2 (2).4
IWTGAS
Estimate water temperatures and dissolved gas concs
4.2 (2).5
IQUAL
Simulate quality constituents using simple relationships with solids and/or water yield
4.2 (2).6
4.2 (2).6
IPTOT
Place point-valued output in INPAD
4.2(2).7
4.2(2).7
IBAROT
Place bar-valued output in INPAD
4.2(2).8
4.2(2)8
IPRINT
Produce printed output
4.2(2)9
4.2 (2).9

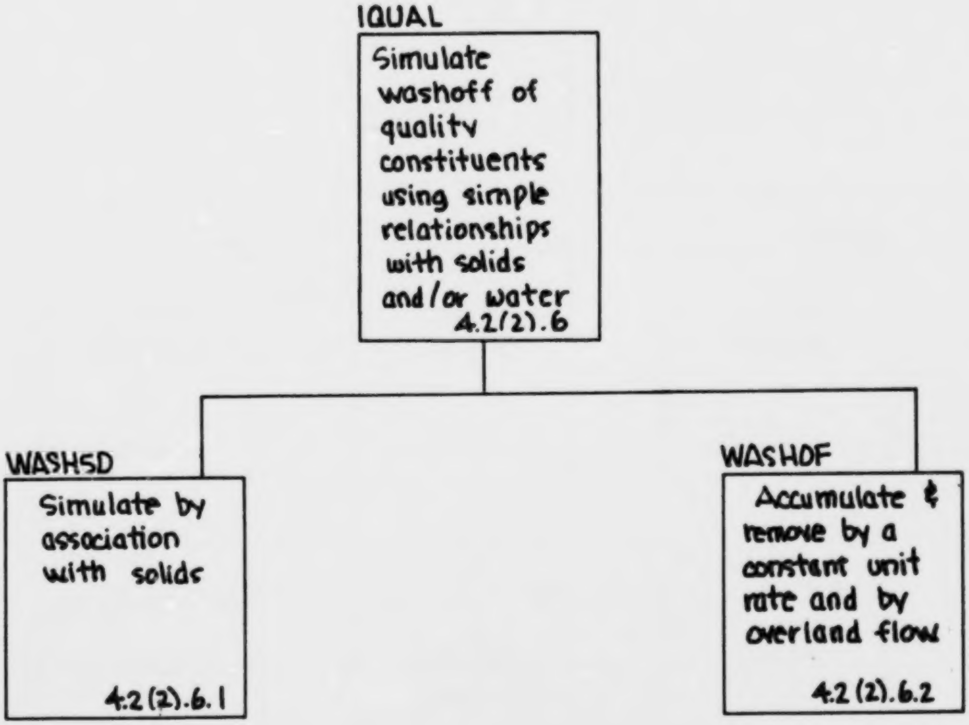
Structure chart 4.2 (2) Impervious land-segment application module


IWATER
Simulate water budget for impervious land segment
4.2 (2).3
RETN
Simulate moisture retention
4.2 (2).3.1
IROUTE
Determine how much of the moisture supply runs off
4.2 (2).3.2
EVRETN
Evaporate from retention storage
4.2 (2).3.3

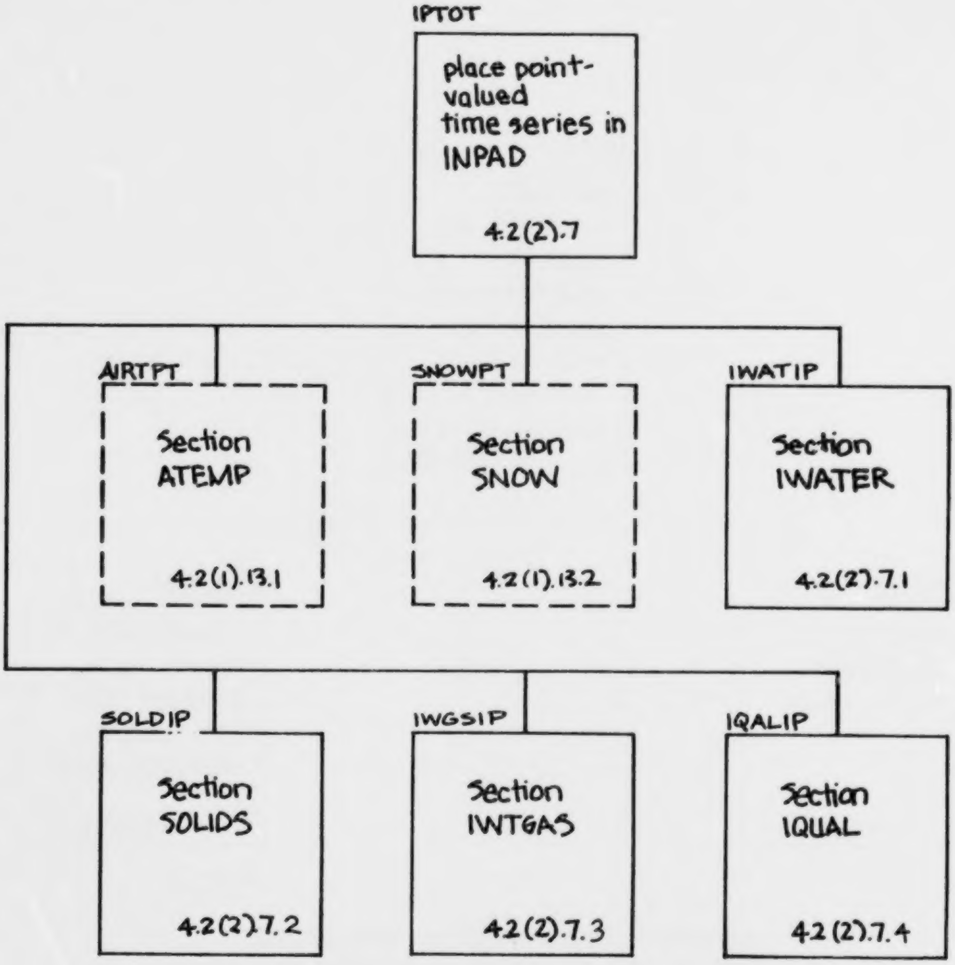
Structure chart 4.2 (2).3 The IWATER section of the IMPLND application module


SOLIDS
Accumulate and remove solids
4.2(2).4
alternative subroutines
SOSLD1
Wash off solids using method 1
4.2(2).4.1
SOSLD2
Wash off solids using method 2
4.2(2).4.2
ACCUM
Accumulate & remove solids independently of runoff
4.2(2).4.3

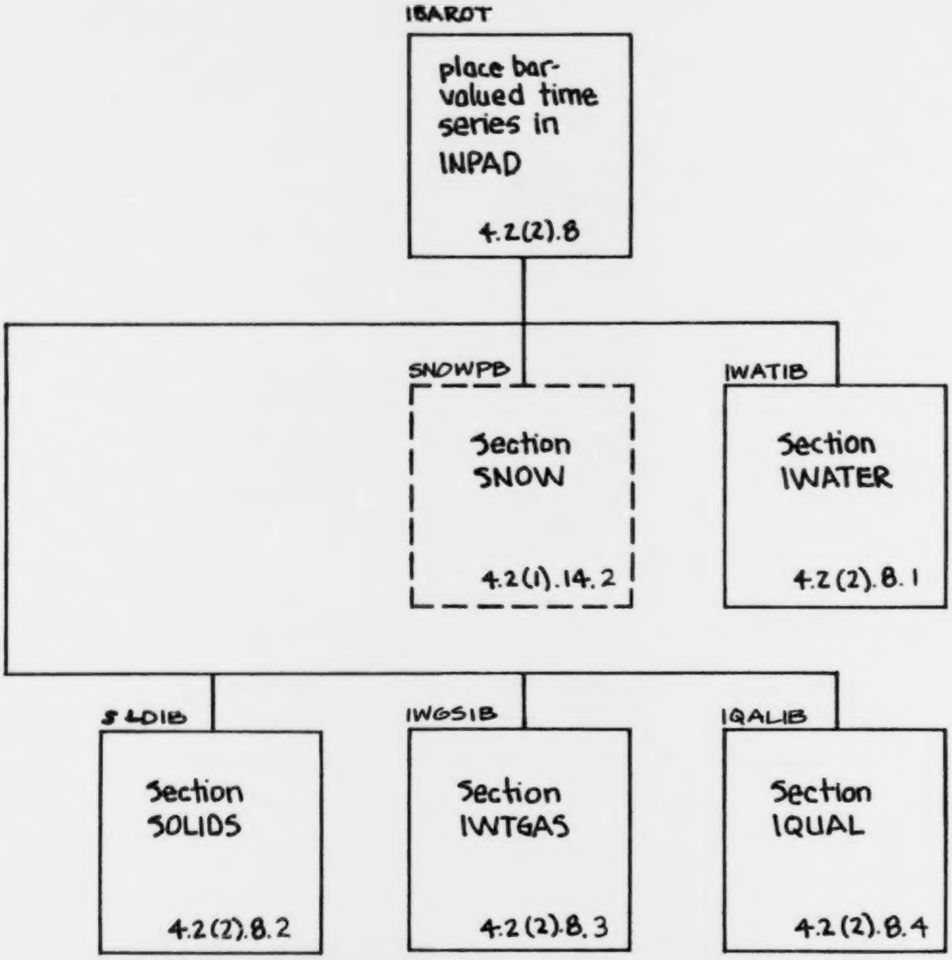
Structure chart 4.2(2).4 The SOLIDS section of the IMPLND application module


IQUAL
Simulate washoff of quality constituents using simple relationships with solids and/or water
4.2(2).6
WASHSD
Simulate by association with solids
4.2(2).6.1
WASHOF
Accumulate & remove by a constant unit rate and by overland flow
4.2(2).6.2

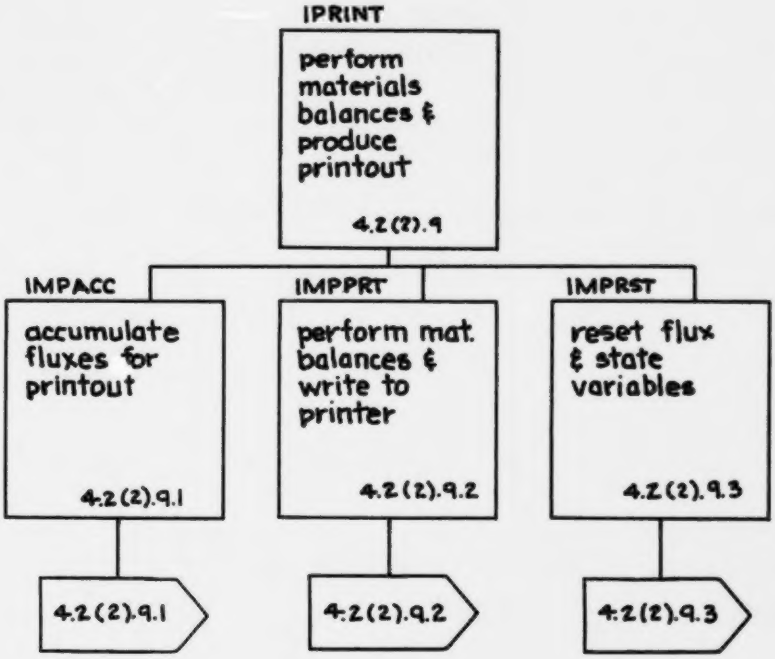
Structure chart 4.2(2).6 The IQUAL section of the IMPLND application module



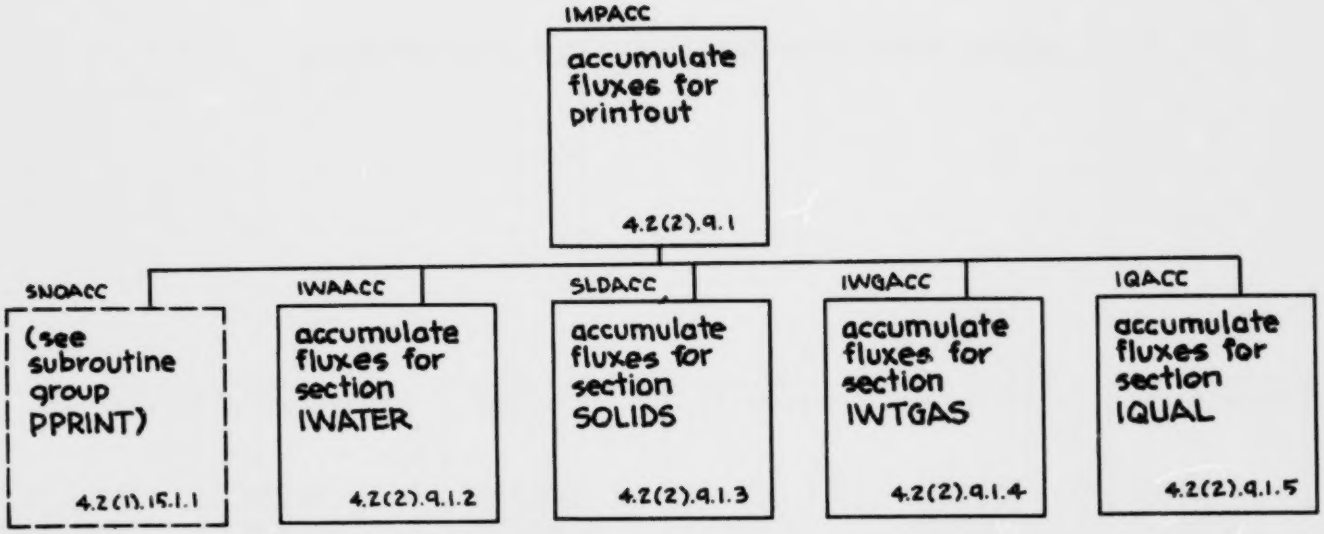
Structure chart 4.2(2).7 Subroutine group IPTOT



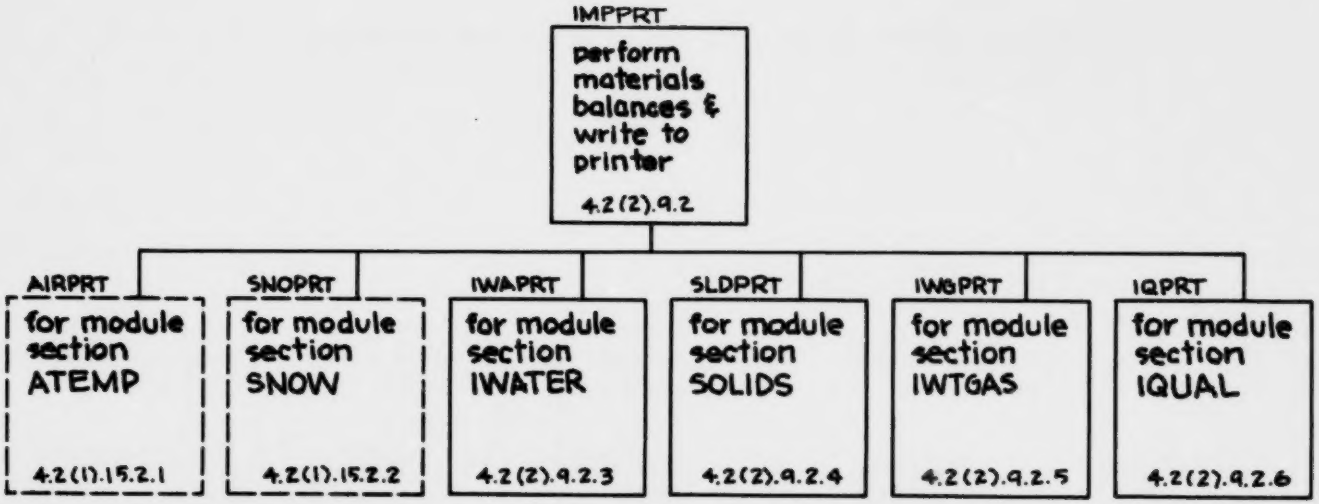
Structure chart 4.2(2).8 Subroutine group IBAROT


IPRINT
perform materials balances & produce printout
4.2(2).9
IMPACC
accumulate fluxes for printout
4.2(2).9.1
IMPPRT
perform mat. balances & write to printer
4.2(2).9.2
IMPRST
reset flux & state variables
4.2(2).9.3
4.2(2).9.1
4.2(2).9.2
4.2(2).9.3

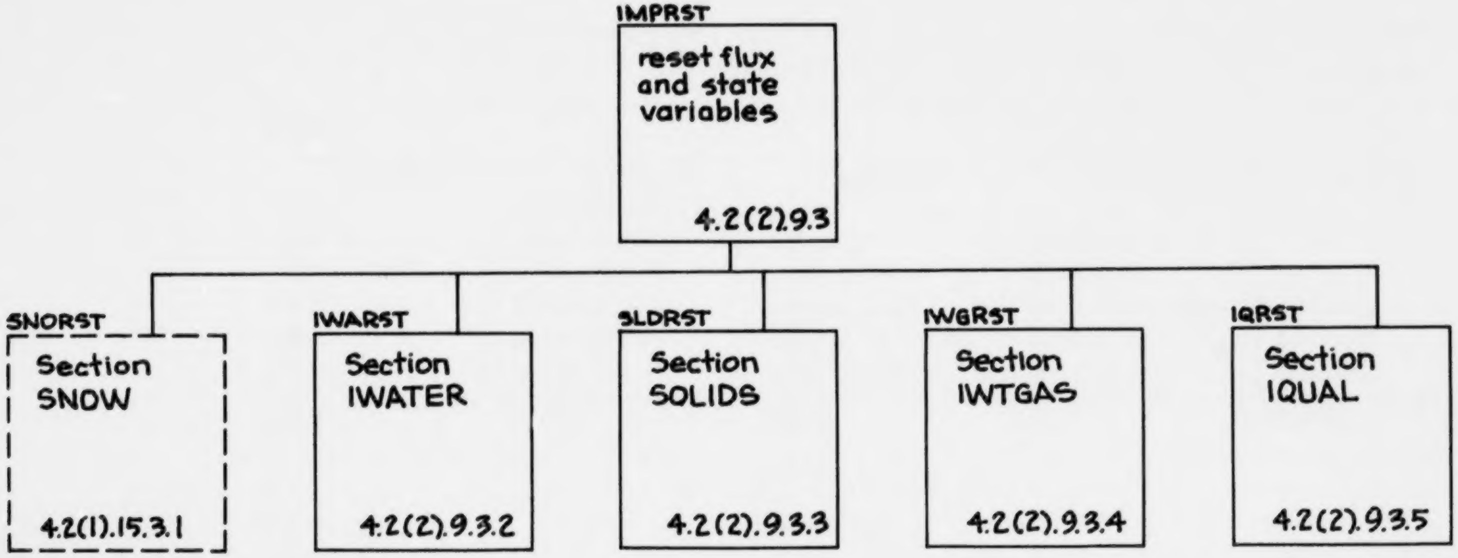
Structure chart 4.2(2).9 Subroutine group IPRINT


IMPACC
accumulate fluxes for printout
4.2(2).9.1
SNOACC
(see subroutine group PPRINT)
4.2(1).15.1.1
IWAACC
accumulate fluxes for section IWATER
4.2(2).9.1.2
SLDACC
accumulate fluxes for section SOLIDS
4.2(2).9.1.3
IWGACC
accumulate fluxes for section IWTGAS
4.2(2).9.1.4
IQACC
accumulate fluxes for section IQUAL
4.2(2).9.1.5

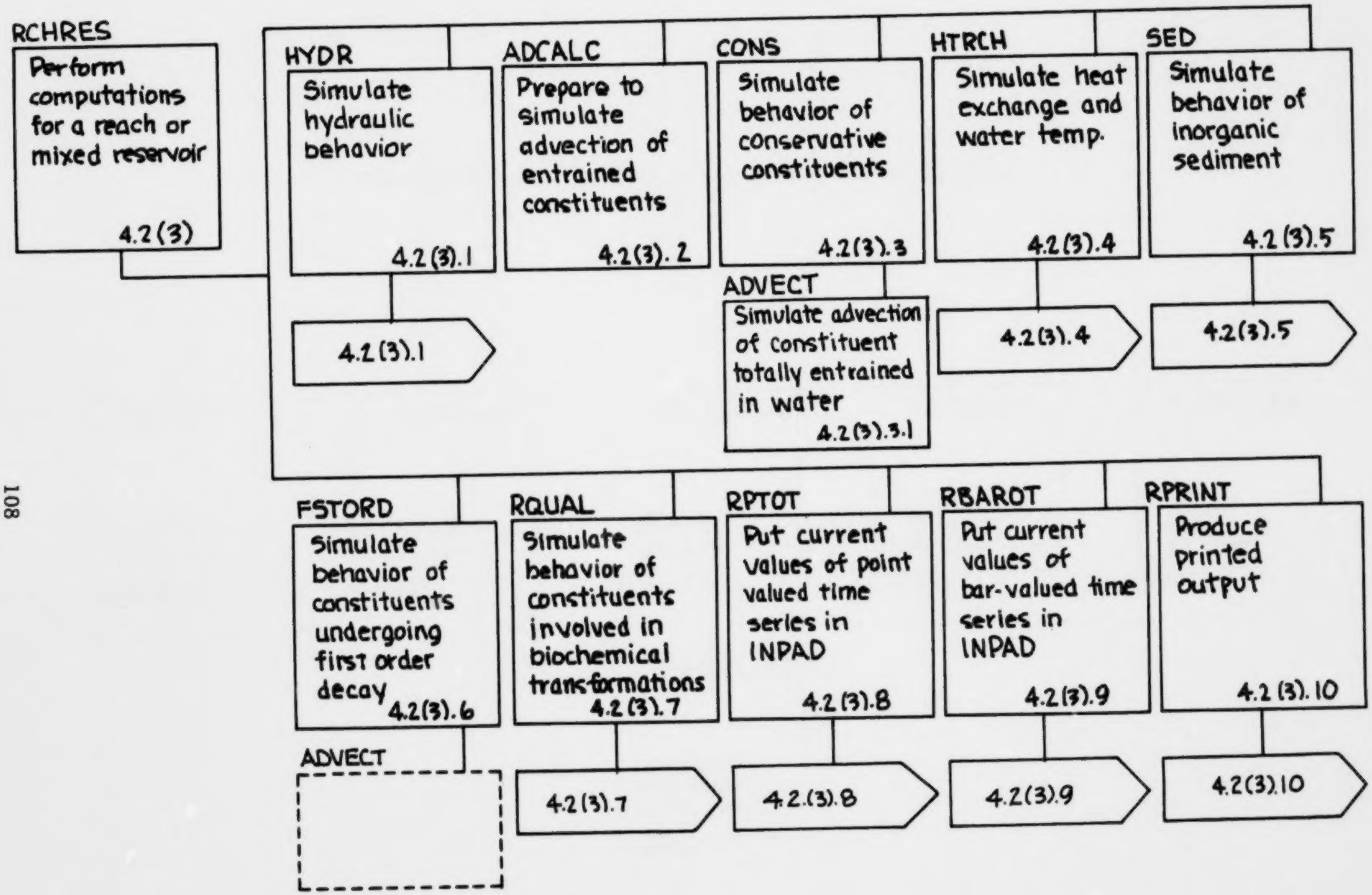
Structure chart 4.2(2).9.1 Subroutine group IMPACC



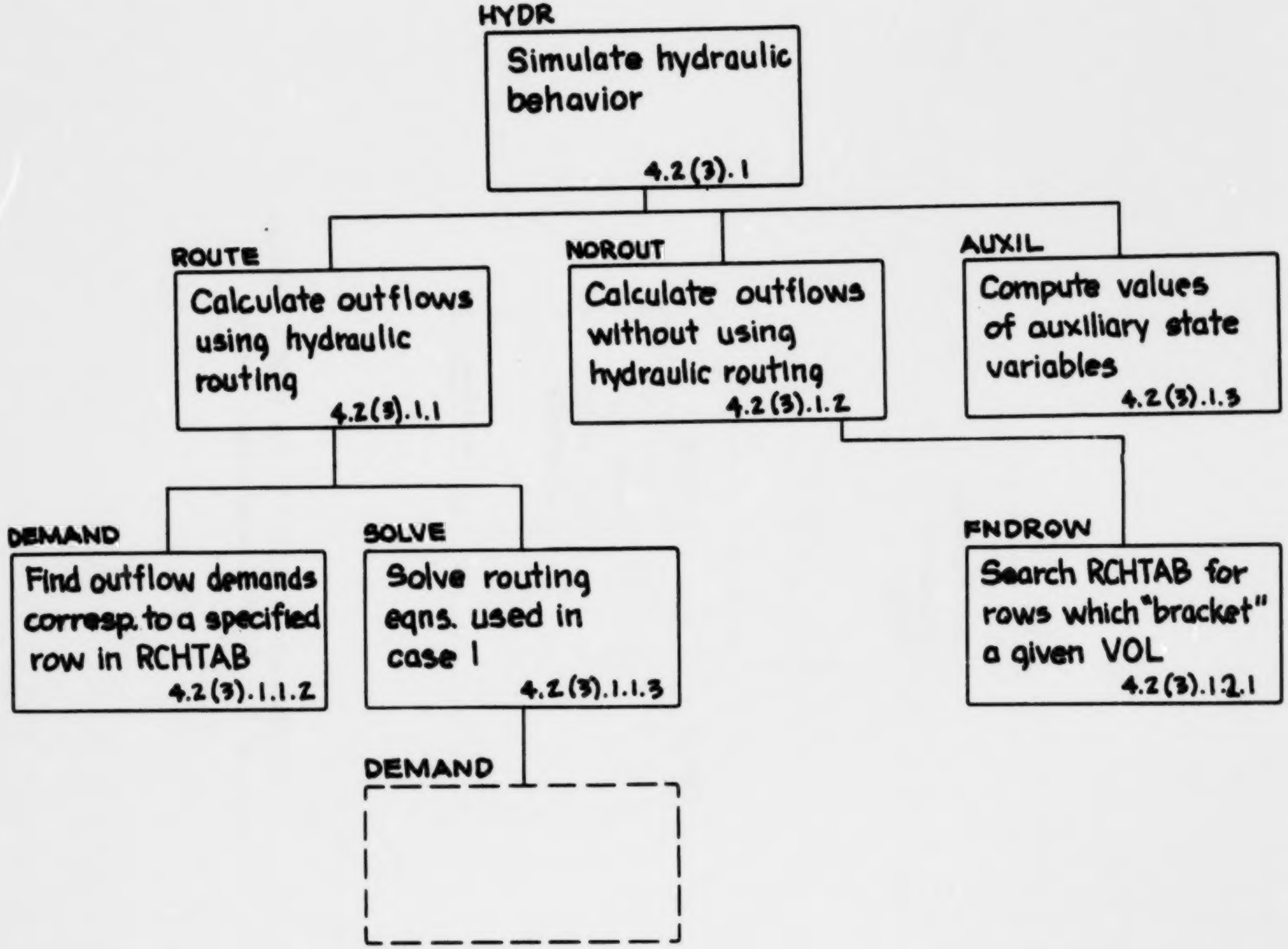
Structure chart 4.2(2).9.2 Subroutine group IMPPRT



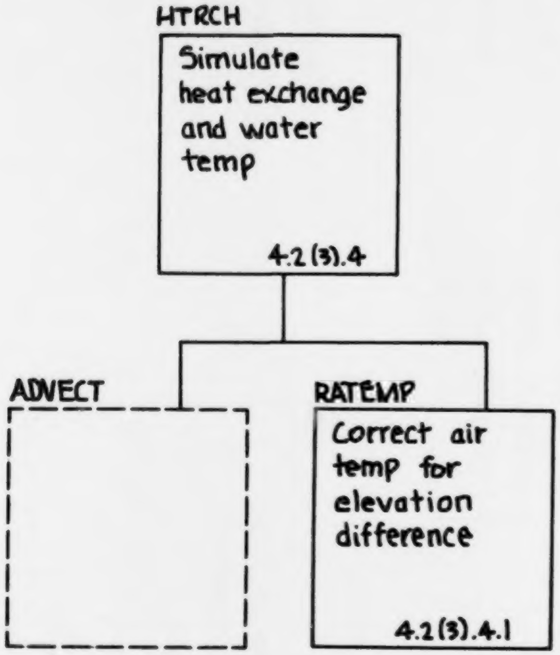
Structure chart 4.2(2).9.3 Subroutine group IMPRST


RCHRES
Perform computations for a reach or mixed reservoir
4.2(3)
HYDR
Simulate hydraulic behavior
4.2(3).1
4.2(3).1
ADCALC
Prepare to simulate advection of entrained constituents
4.2(3).2
CONS
Simulate behavior of conservative constituents
4.2(3).3
ADVECT
Simulate advection of constituent totally entrained in water
4.2(3).3.1
HTRCH
Simulate heat exchange and water temp.
4.2(3).4
4.2(3).4
SED
Simulate behavior of inorganic sediment
4.2(3).5
4.2(3).5
FSTORD
Simulate behavior of constituents undergoing first order decay
4.2(3).6
ADVECT
RQUAL
Simulate behavior of constituents involved in biochemical transformations
4.2(3).7
4.2(3).7
RPTOT
Put current values of point valued time series in INPAD
4.2(3).8
4.2.(3).8
RBAROT
Put current values of bar-valued time series in INPAD
4.2(3).9
4.2(3).9
RPRINT
Produce printed output
4.2(3).10
4.2(3).10

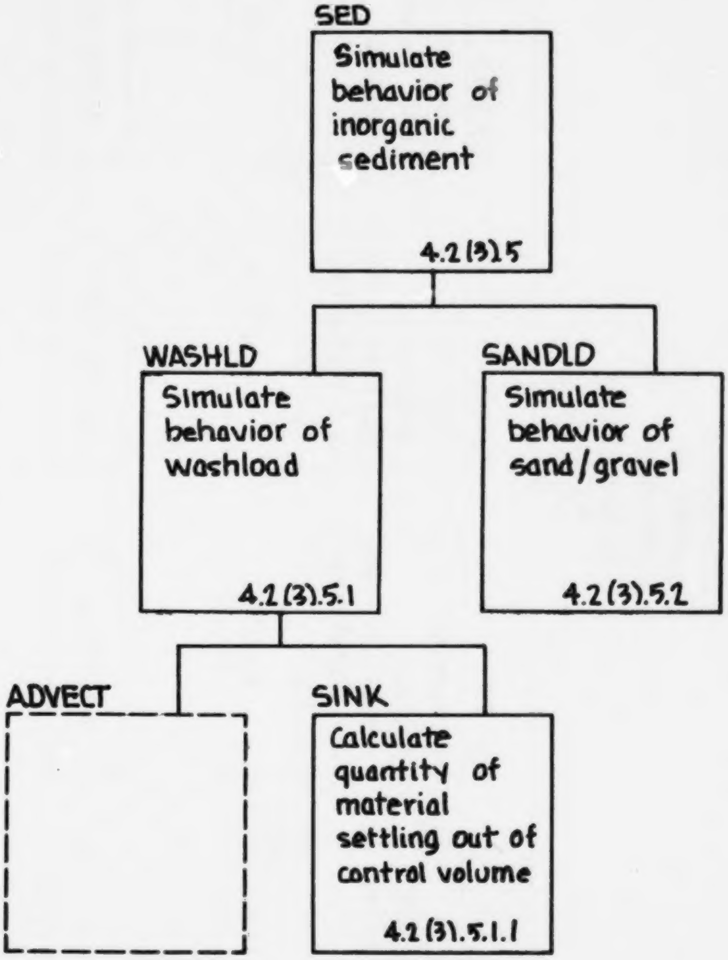
Structure chart 4.2(3) Reach/mixed reservoir application module.


HYDR
Simulate hydraulic behavior
4.2(3).1
ROUTE
Calculate outflows using hydraulic routing
4.2(3).1.1
NOROUT
Calculate outflows without using hydraulic routing
4.2(3).1.2
AUXIL
Compute values of auxiliary state variables
4.2(3).1.3
DEMAND
Find outflow demands corresp. to a specified row in RCHTAB
4.2(3).1.1.2
SOLVE
Solve routing eqns. used in case 1
4.2(3).1.1.3
FNDROW
Search RCHTAB for rows which "bracket" a given VOL
4.2(3).1.2.1
DEMAND

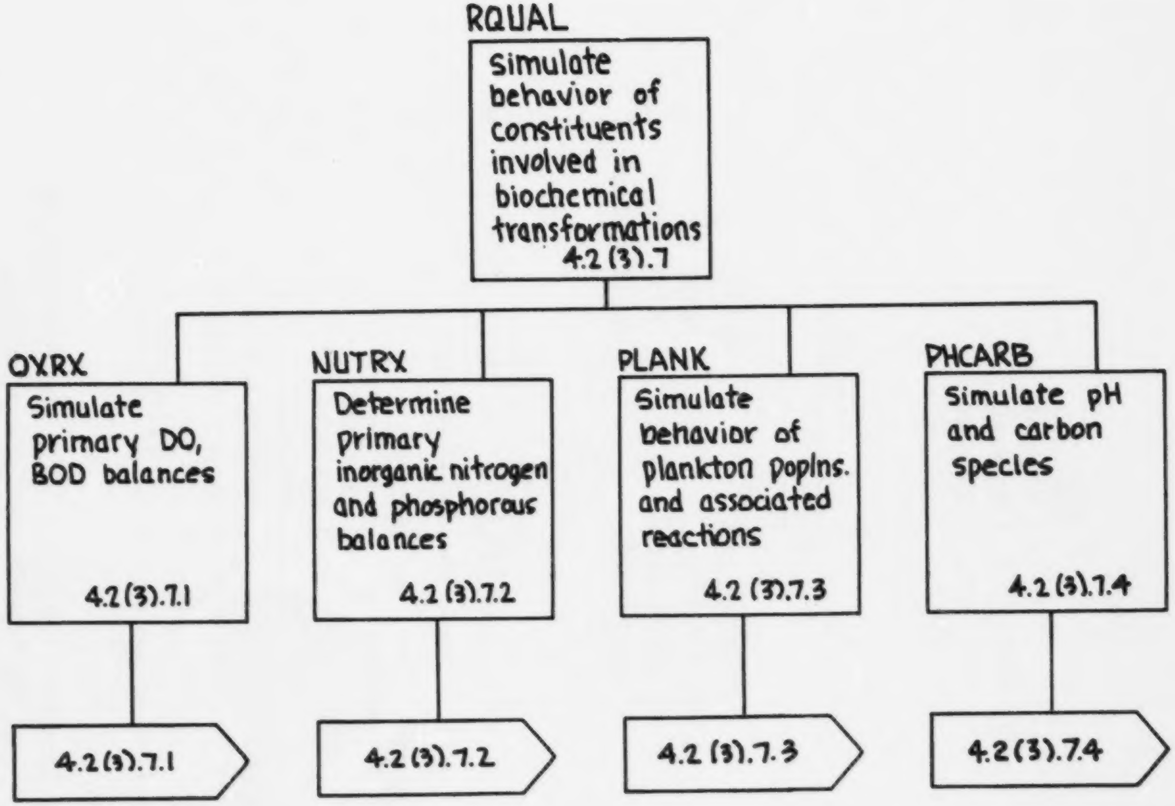
Structure chart 4.2(3).1 The HYDR section of the RCHRES application module


HTRCH
Simulate heat exchange and water temp
4.2 (3).4
ADVECT
RATEMP
Correct air temp for elevation difference
4.2 (3).4.1

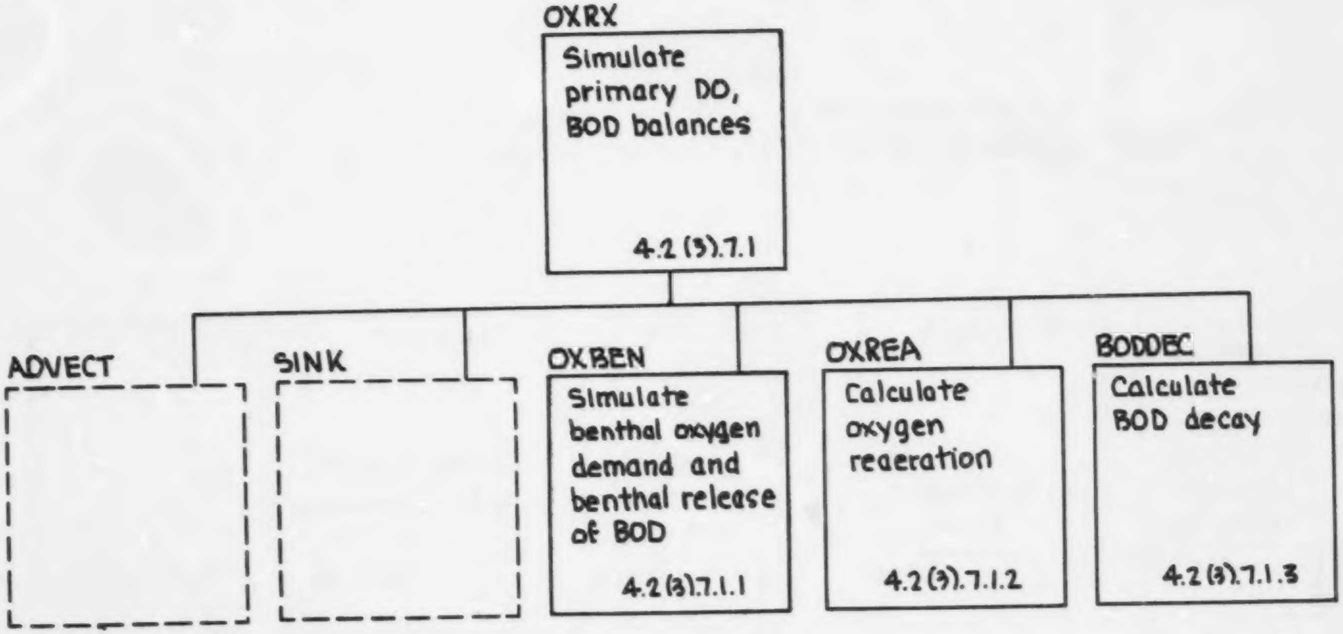
Structure chart 4.2 (3).4 The HTRCH section of the RCHRES application module


SED
Simulate behavior of inorganic sediment
4.2(3).5
WASHLD
Simulate behavior of washload
4.2(3).5.1
SANDLD
Simulate behavior of sand/gravel
4.2(3).5.2
ADVECT
SINK
Calculate quantity of material settling out of control volume
4.2(3).5.1.1

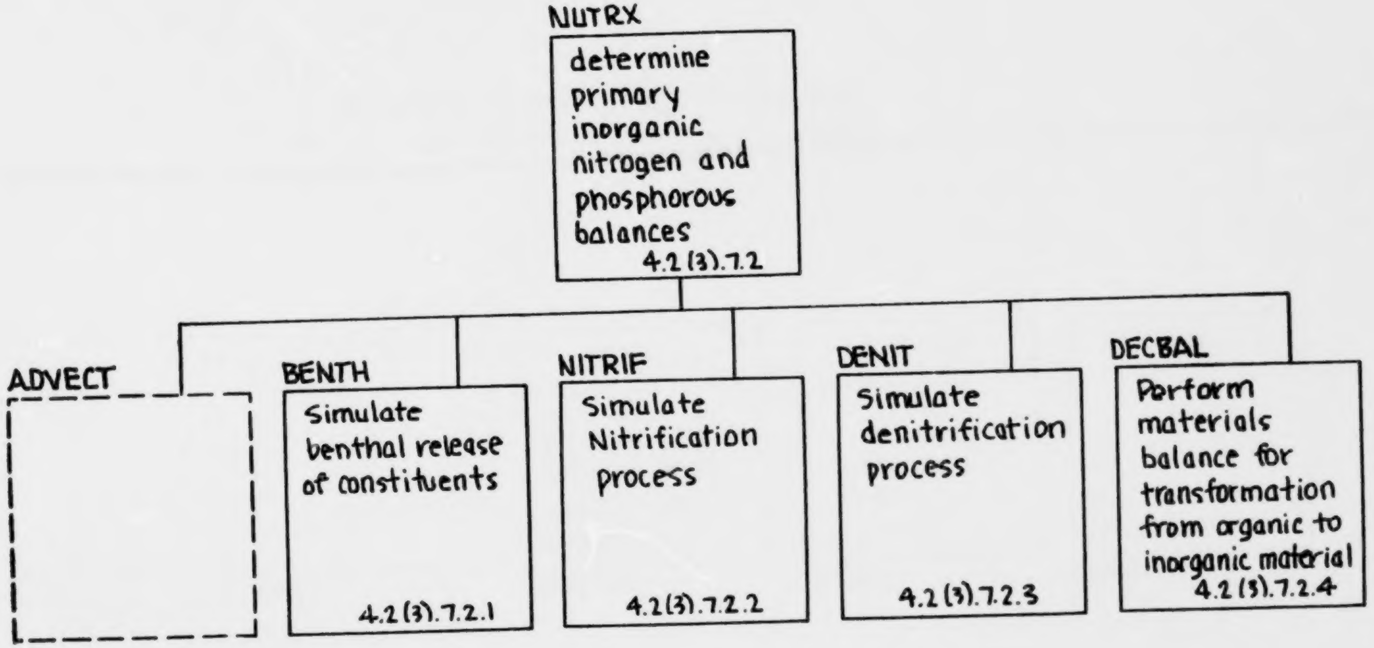
Structure chart 4.2(3).5 The SED section of the RCHRES application module


RQUAL
Simulate behavior of constituents involved in biochemical transformations
4.2 (3).7
OXRX
Simulate primary DO, BOD balances
4.2 (3).7.1
NUTRX
Determine primary inorganic nitrogen and phosphorous balances
4.2 (3).7.2
PLANK
Simulate behavior of plankton poplns. and associated reactions
4.2 (3).7.3
PHCARB
Simulate pH and carbon species
4.2 (3).7.4
4.2 (3).7.1
4.2 (3).7.2
4.2 (3).7.3
4.2 (3).7.4

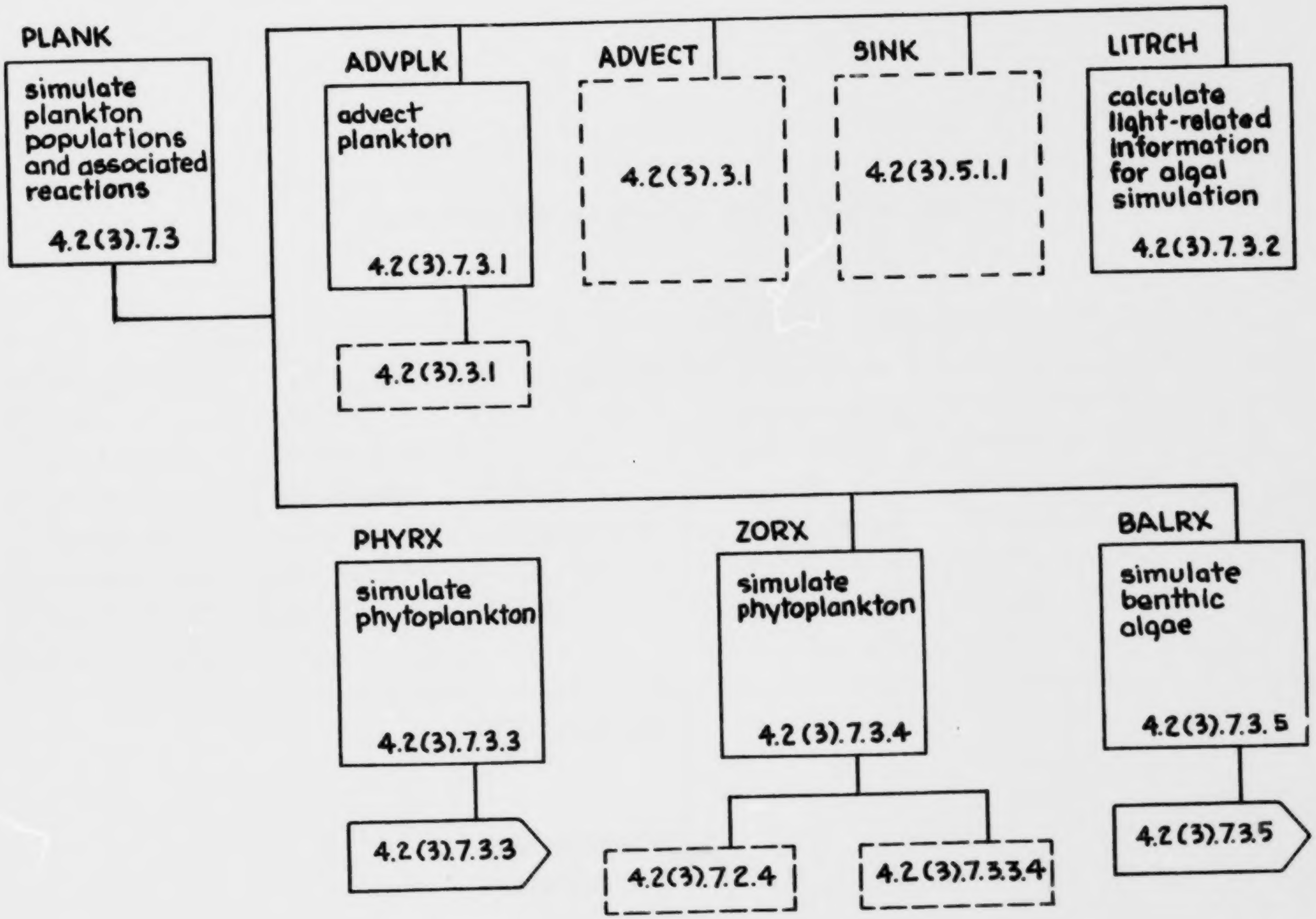
Structure chart 4.2 (3).7 The RQUAL section of the RCHRES application module


OXRX
Simulate primary DO, BOD balances
4.2 (3).7.1
ADVECT
SINK
OXBEN
Simulate benthal oxygen demand and benthal release of BOD
4.2 (3).7.1.1
OXREA
Calculate oxygen reaeration
4.2 (3).7.1.2
BODDEC
Calculate BOD decay
4.2 (3).7.1.3

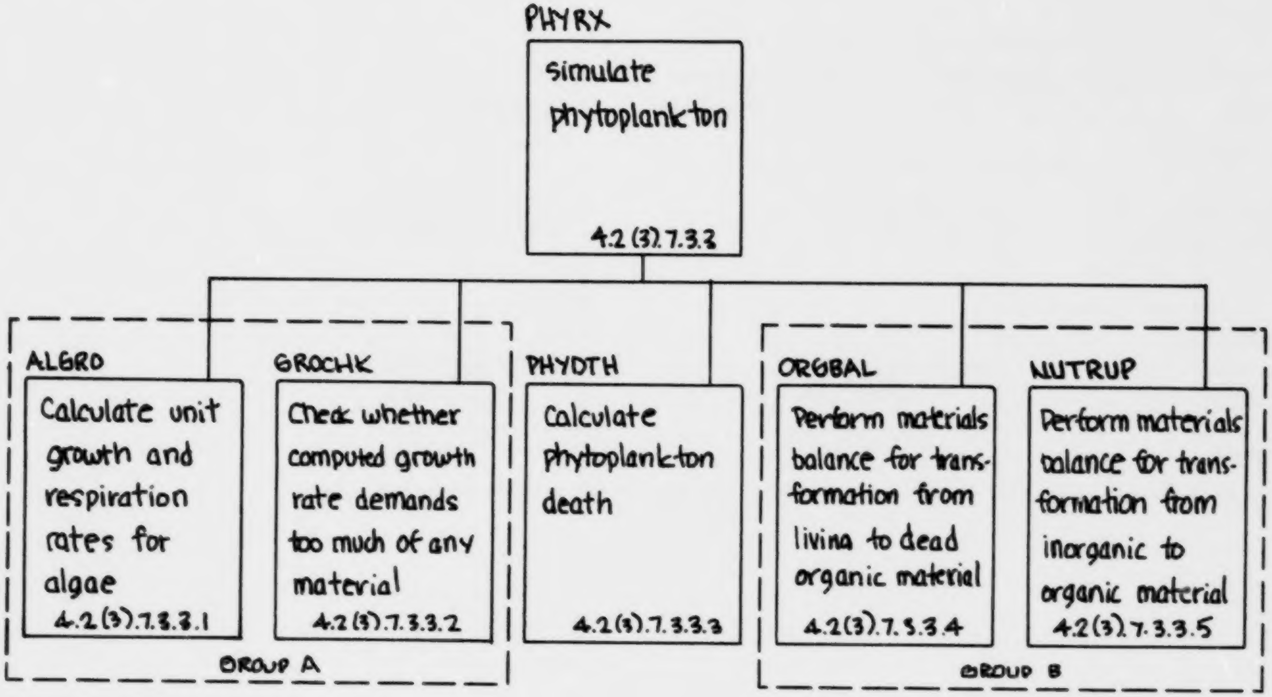
Structure chart 4.2 (3).7.1 OXRX
subroutine group. DO, BOD simulation


NUTRX
determine primary inorganic nitrogen and phosphorous balances
4.2 (3).7.2
ADVECT
BENTH
Simulate benthal release of constituents
4.2 (3).7.2.1
NITRIF
Simulate Nitrification process
4.2(3).7.2.2
DENIT
Simulate denitrification process
4.2 (3).7.2.3
DECBAL
Perform materials balance for transformation from organic to inorganic material
4.2 (3).7.2.4

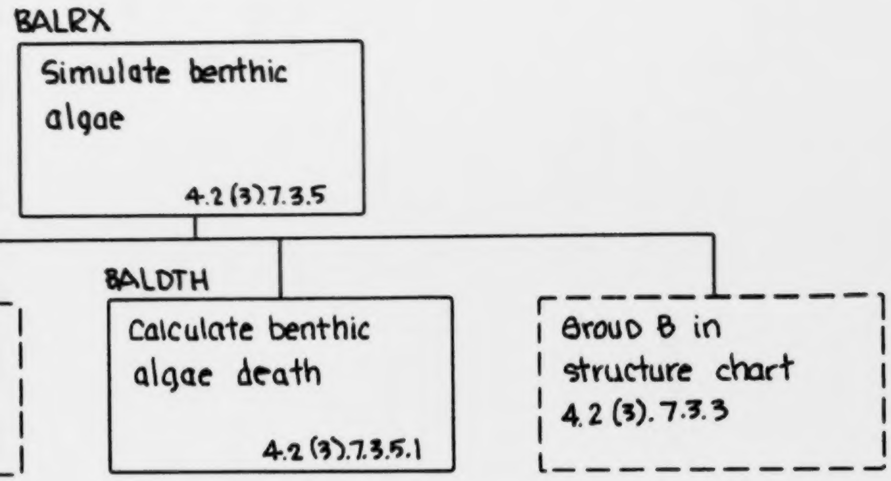
Structure chart 4.2 (3).7.2 NUTRX subroutine group
Primary inorganic N and P simulation



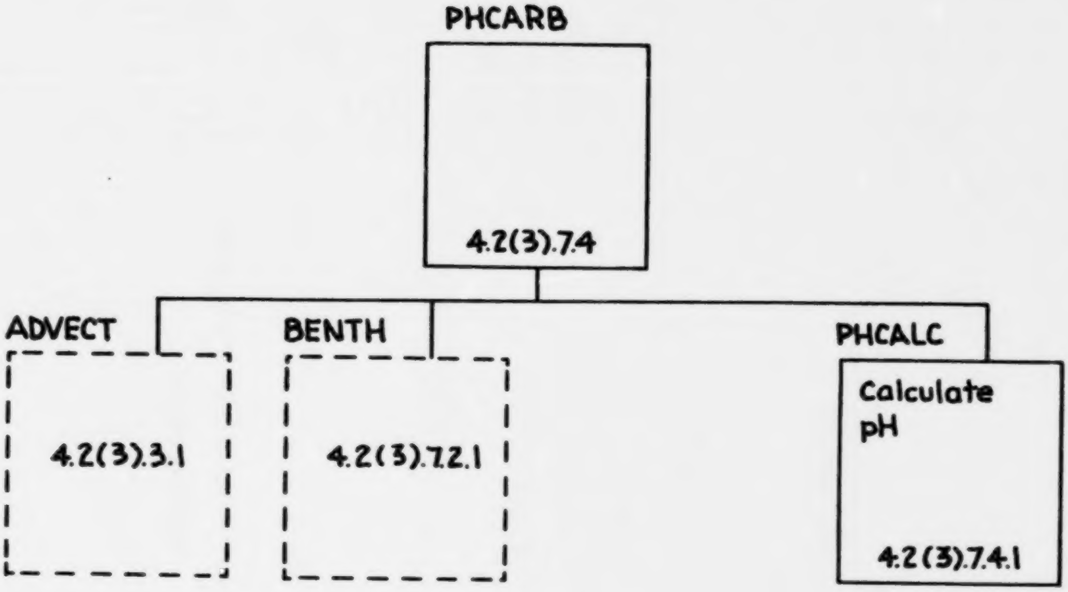
Structure chart 4.2(3).7.3 Simulate plankton populations and associated reactions


PHYRX
Simulate phytoplankton
4.2 (3).7.3.3
ALGRO
Calculate unit growth and respiration rates for algae
4.2 (3).7.3.3.1
GROCHK
Check whether computed growth rate demands too much of any material
4.2 (3).7.3.3.2
GROUP A
PHYDTH
Calculate phytoplankton death
4.2 (3).7.3.3.3
ORGBAL
Perform materials balance for transformation from living to dead organic material
4.2 (3).7.3.3.4
NUTRUP
Perform materials balance for transformation from inorganic to organic material
4.2 (3).7.3.3.5
GROUP B

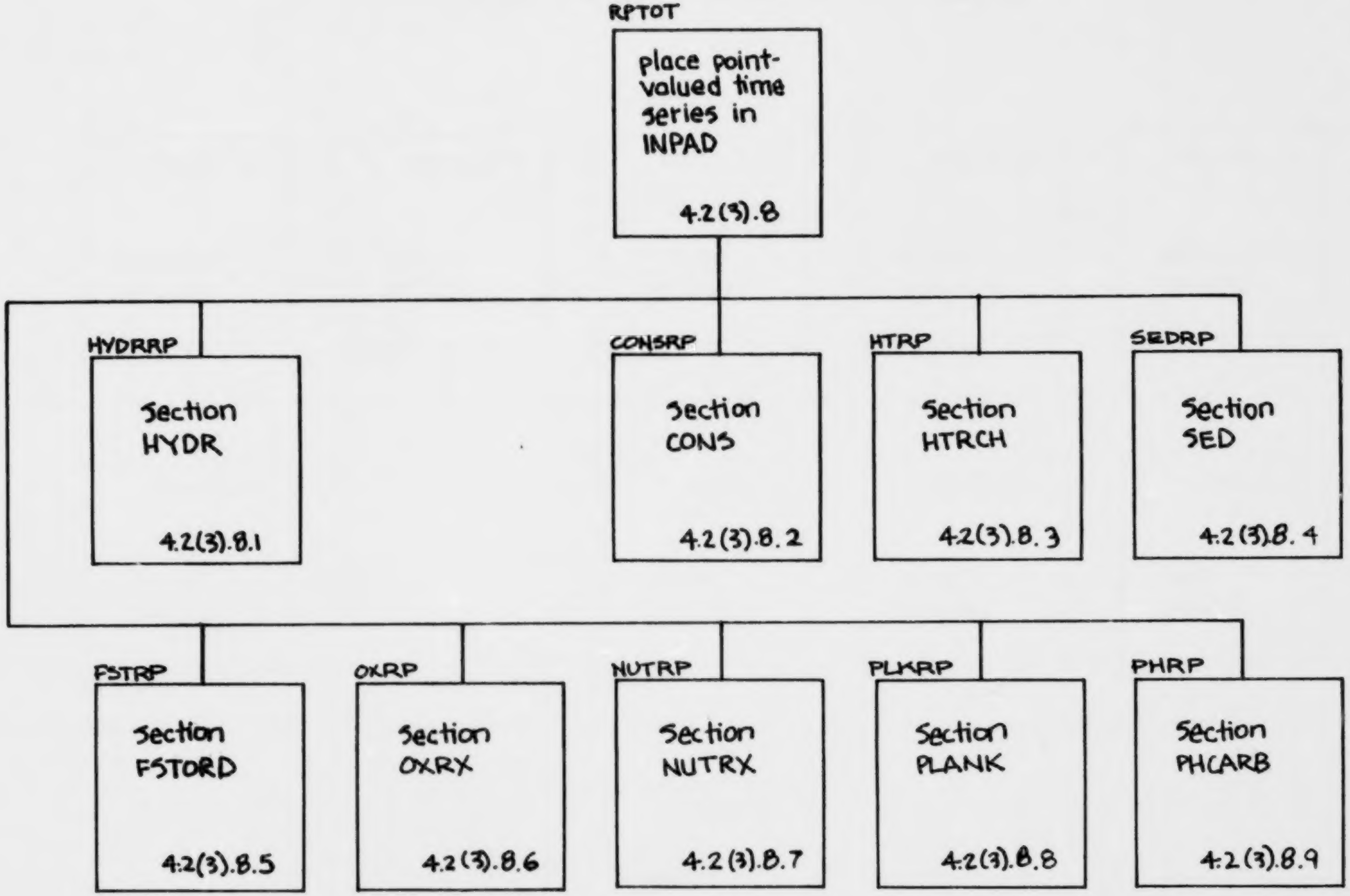
Structure chart 4.2 (3).7.3.3. Simulate phytoplankton


BALRX
Simulate benthic algae
4.2 (3).7.3.5
Group A in structure chart 4.2 (3).7.3.3
BALDTH
Calculate benthic algae death
4.2 (3).7.3.5.1
Group B in structure chart 4.2 (3). 7.3.3

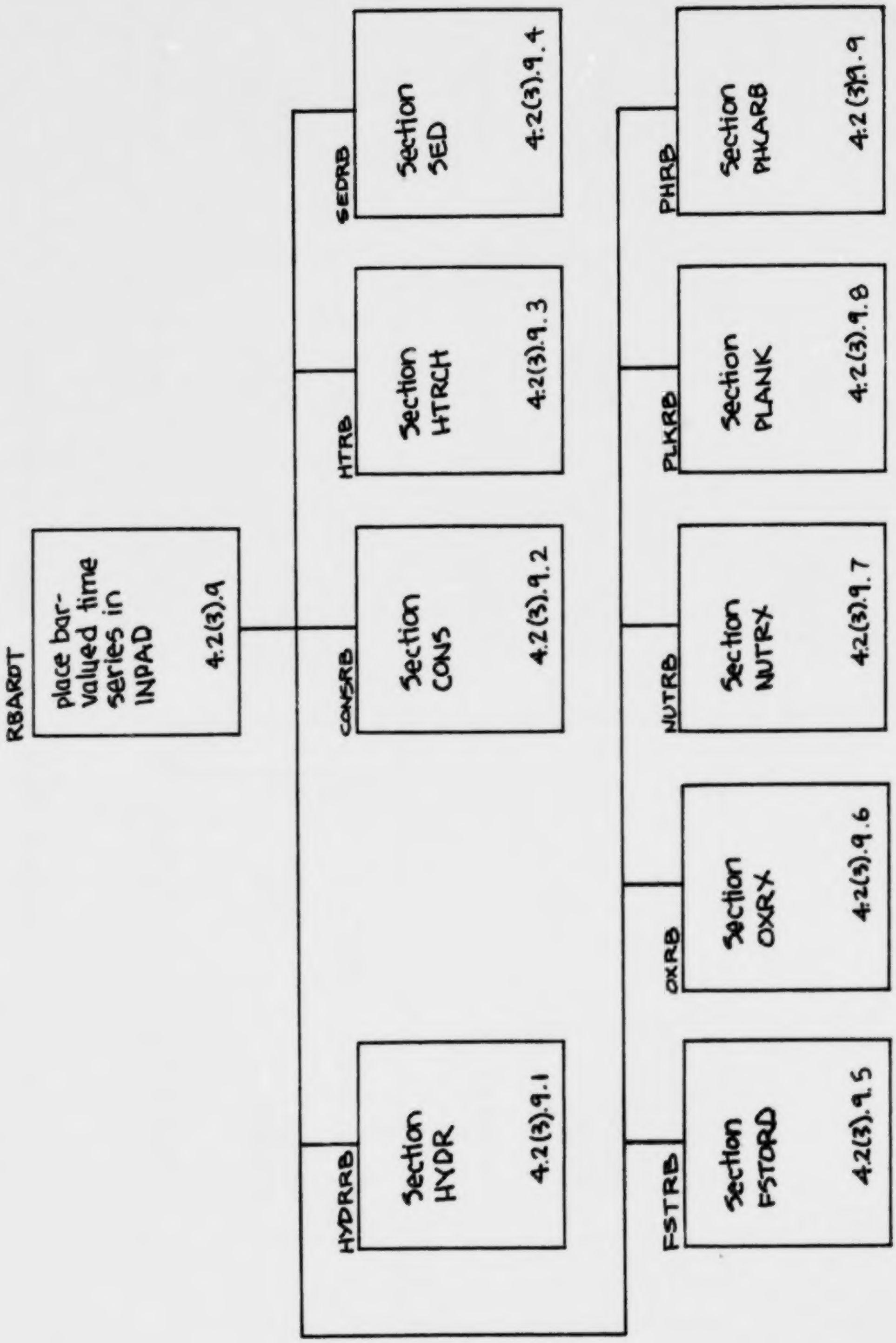
Structure chart 4.2 (3).7.3.5 Simulate benthic algae



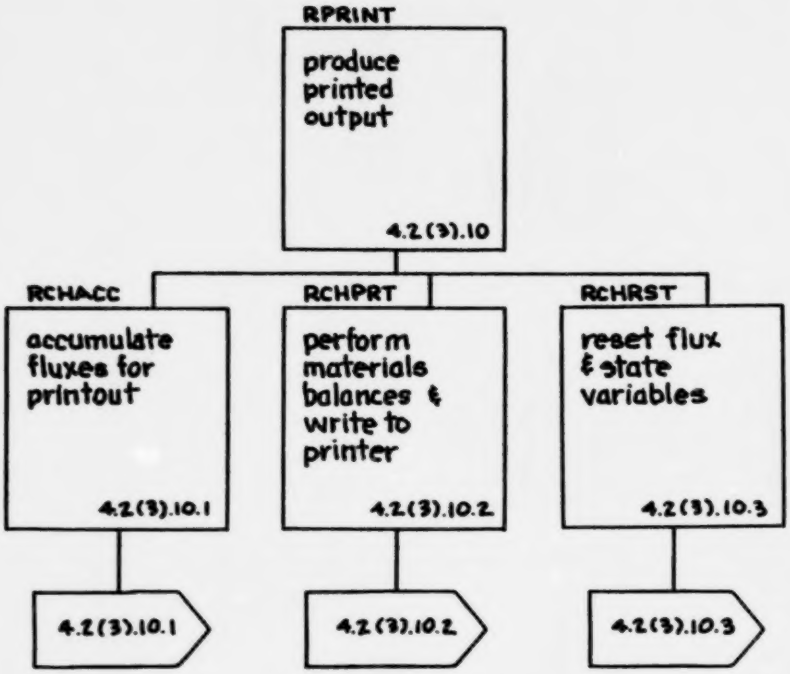
Structure chart 4.2(3).7.4 Simulate pH and carbon species



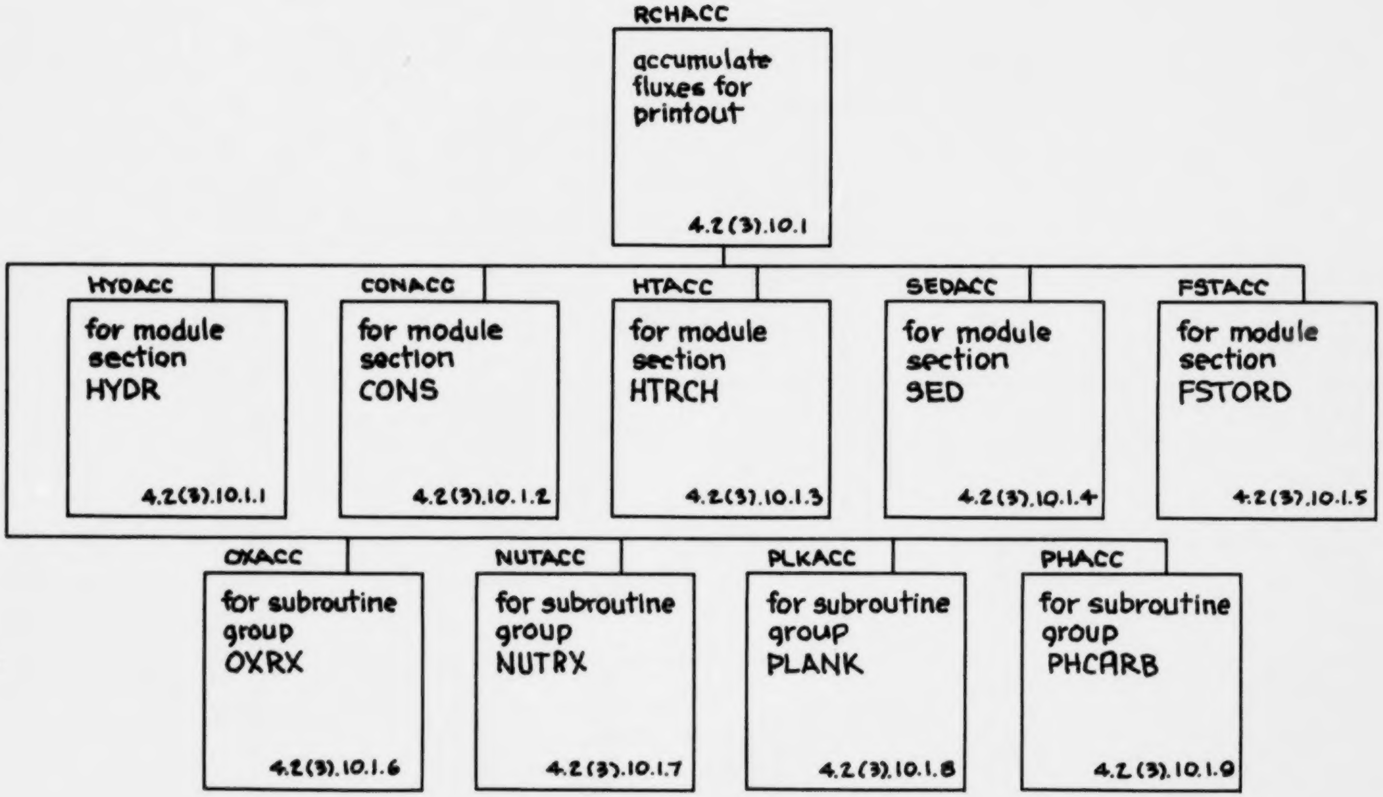
Structure chart 4.2(3).8 Subroutine group RPTOT



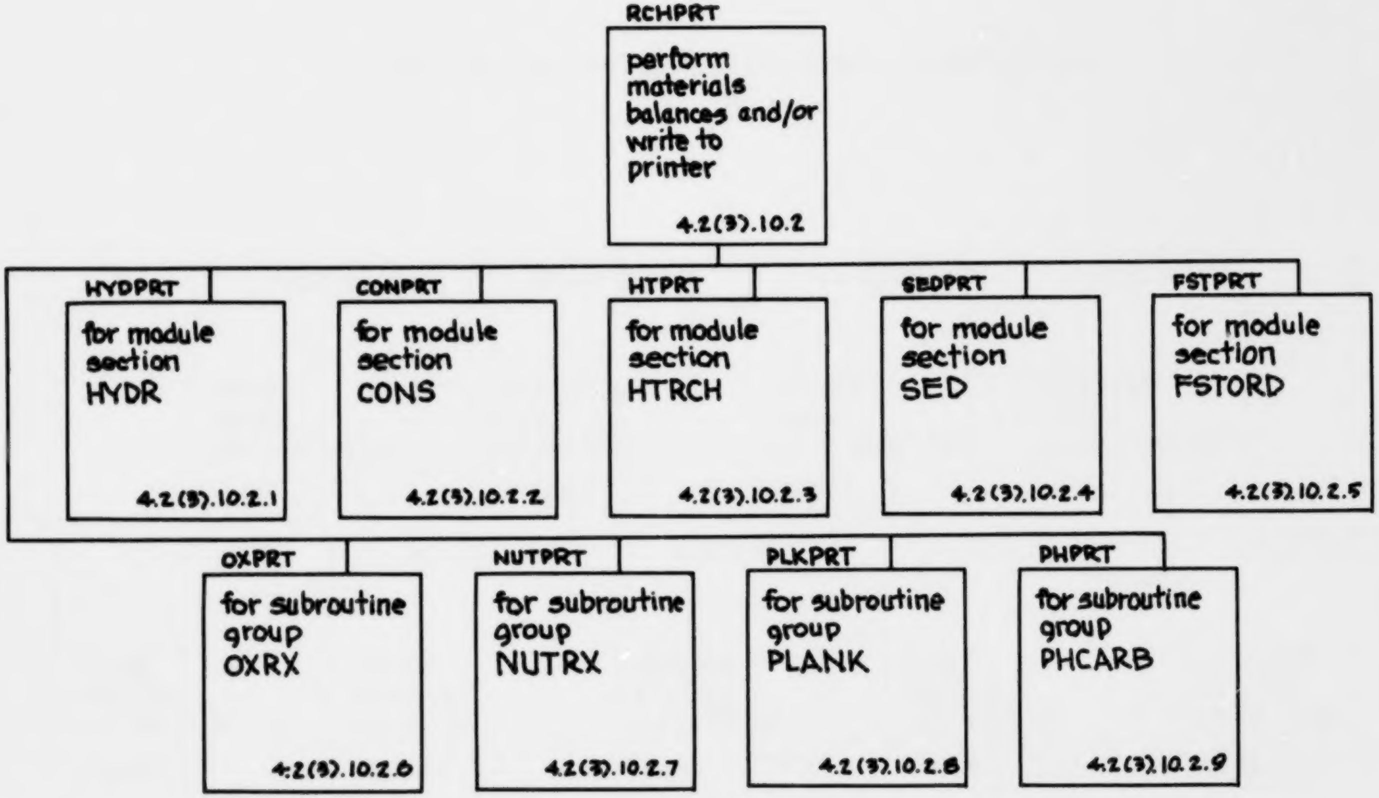
Structure chart 4:2(3).9 Subroutine group RBAROT



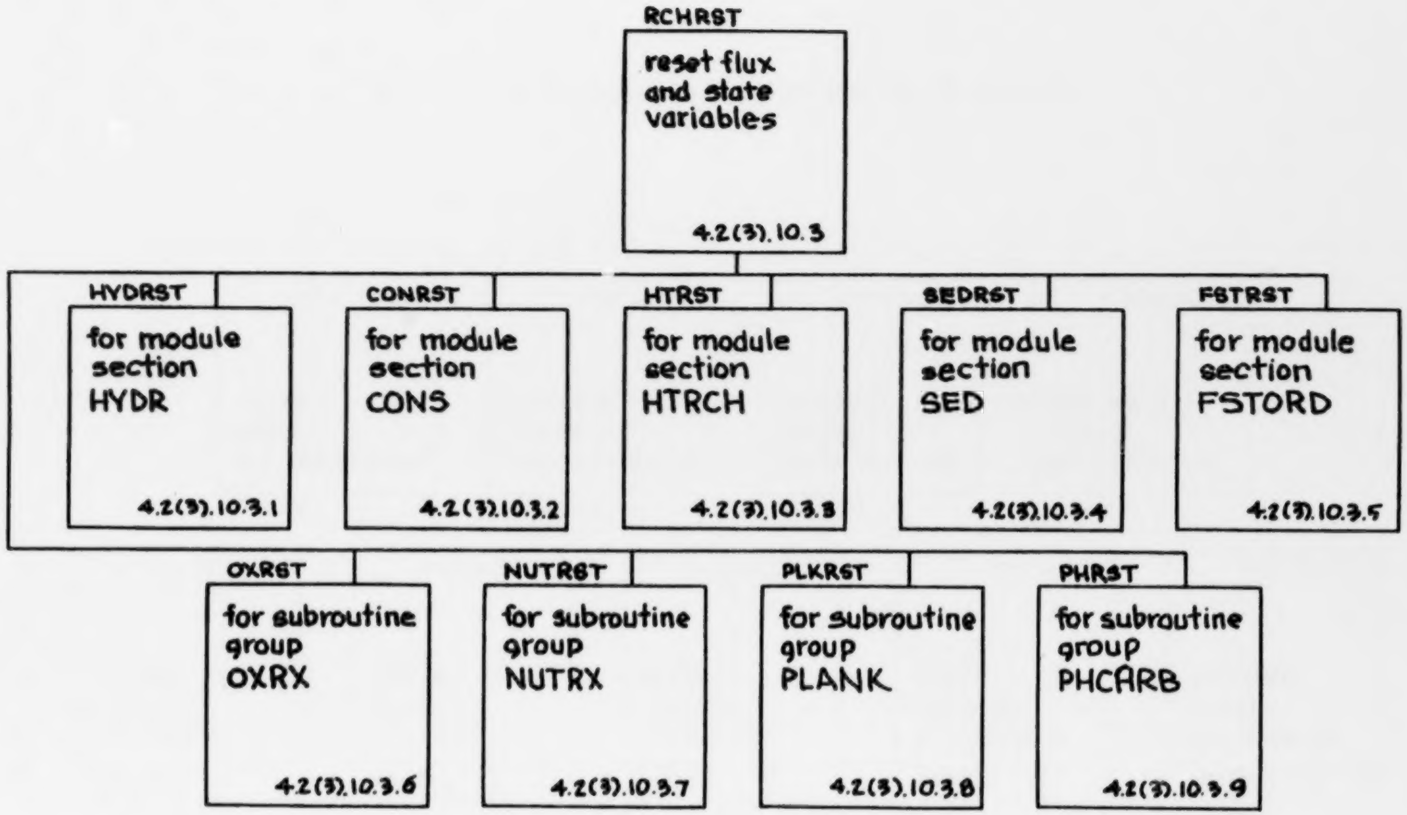
Structure chart 4.2(3).10 The RPRINT section of the RCHRES application module



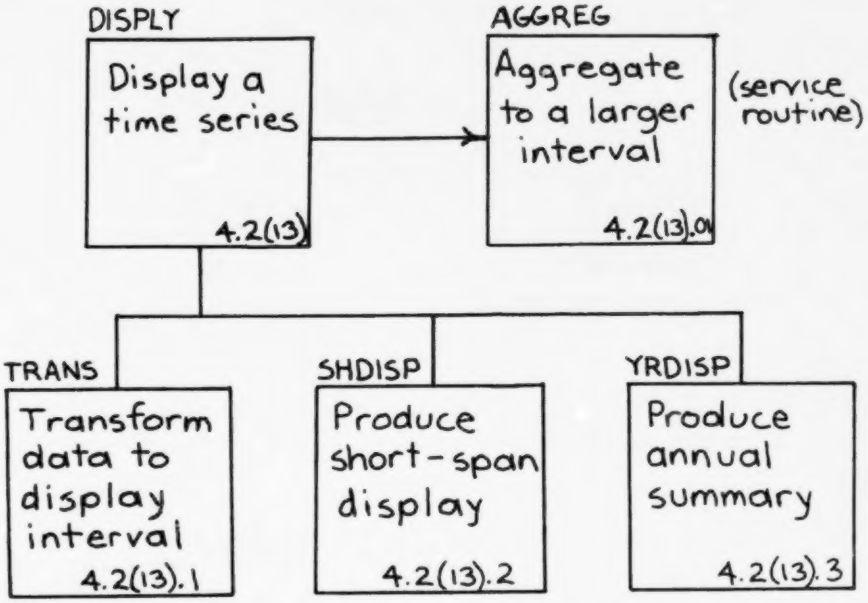
Structure chart 4.2(3).10.1 Subroutine group RCHACC



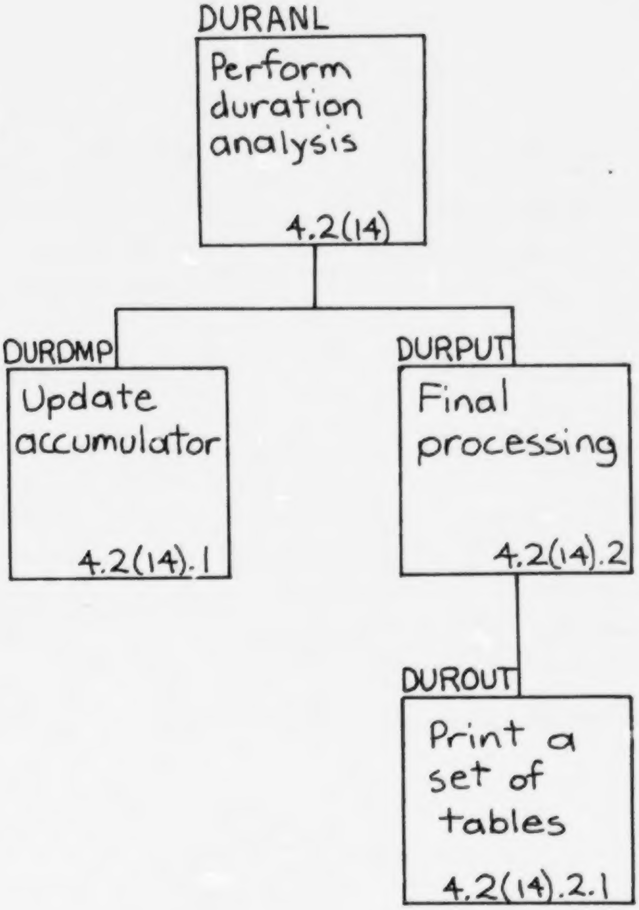
Structure chart 4.2(3).10.2 Subroutine group RCHPRT



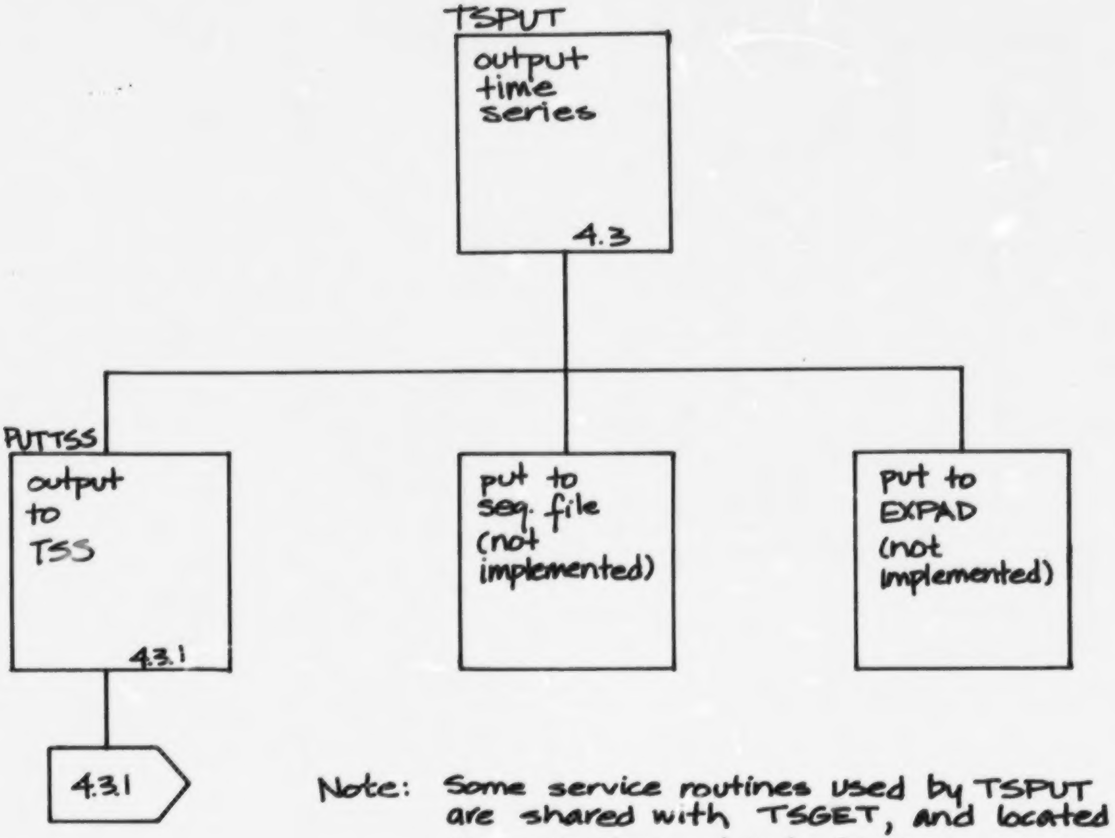
Structure chart 4.2(3).10.3 Subroutine group RCHRST



Structure chart 4.2(13) The Display utility module

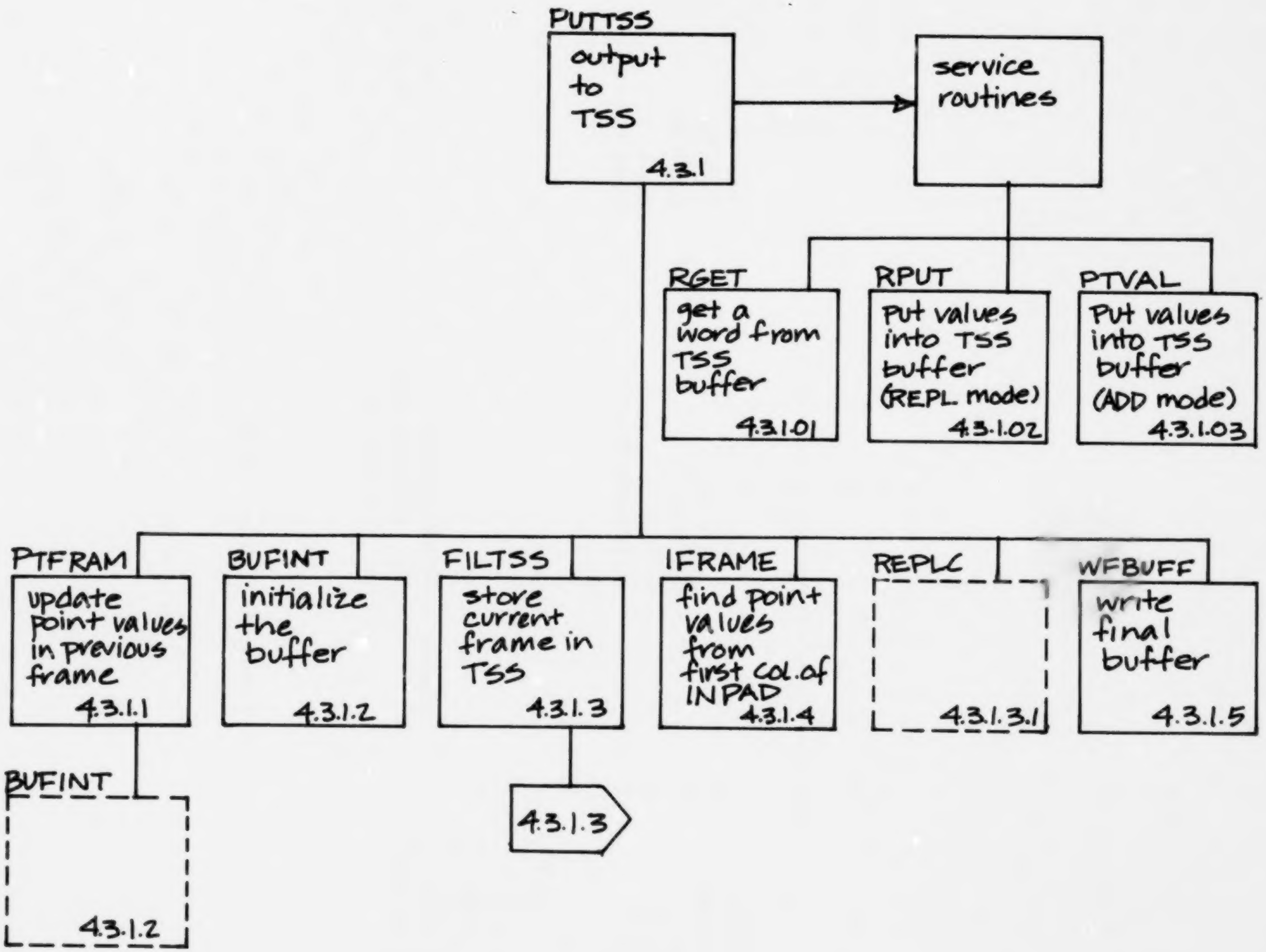


Structure chart 4.2(14) The Duration Analysis utility module

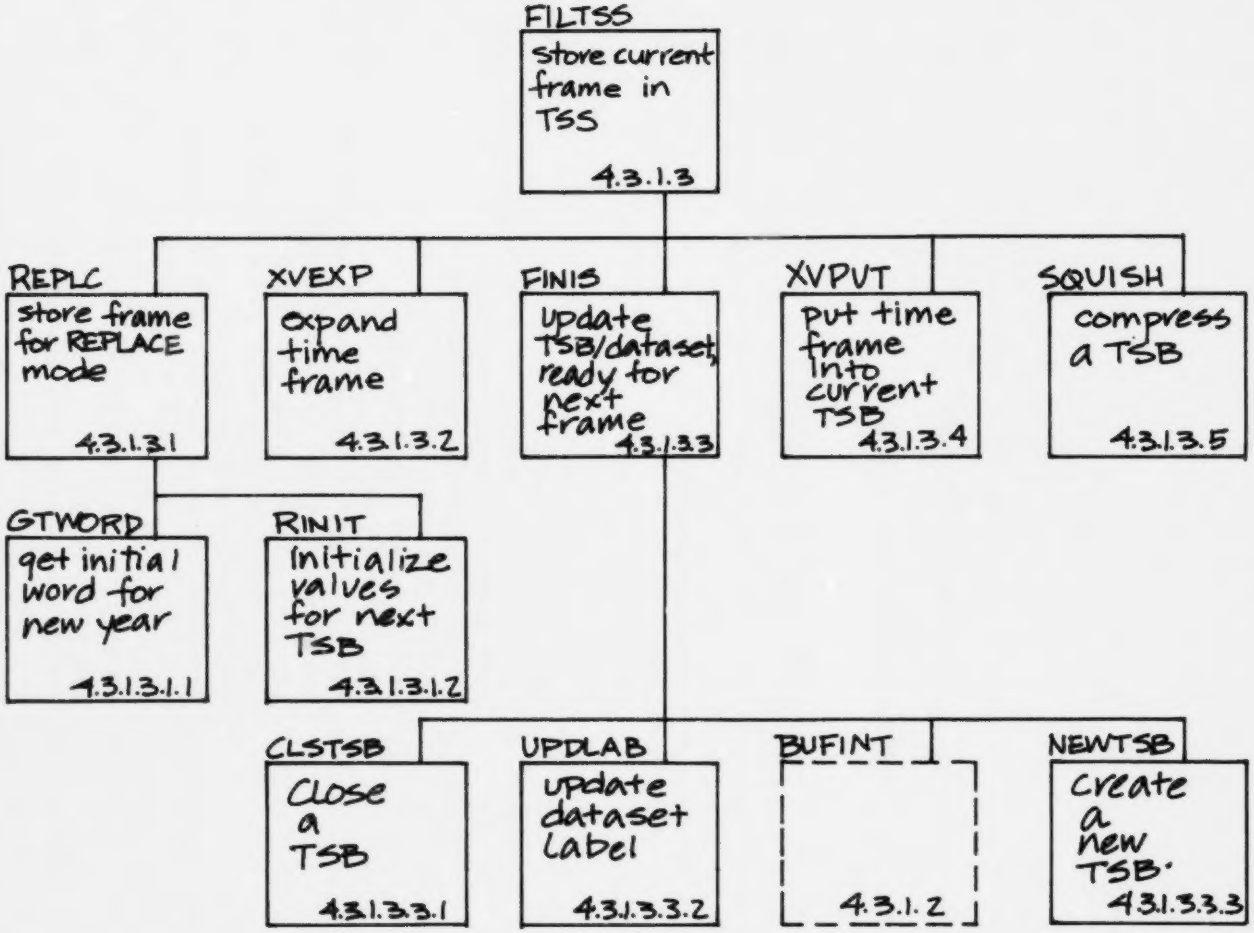

TSPUT
output time series
4.3
PUTTSS
output to TSS
4.3.1
put to seq. file (not implemented)
put to EXPAD (not implemented)
4.3.1

Note: Some service routines used by TSPUT are shared with TSGET, and located in Structure chart 4.1.01

Structure chart 4.3 TSPUT



structure chart 4.3.1 The PUTTSS section of module TSPUT



Structure chart 4.3.1.3 Subroutine group FILTSS

PART E FUNCTIONAL DESCRIPTION

CONTENTS

General Comments . . . .

Descriptions:

1.0 MAIN Program 134

2.0 Manage the Time Series Store (Module TSSMGR) 134

3.0 Interpret a Run Data Set in the User's Control Input (Module INTERP) 135

4.0 Supervise and Perform Operations (Module OSUPER) 137
- 4.03 Perform Special Actions (Subroutine SPECL). 138
- 4.1 Get Required Input Time Series (Module TSGET) 139
- 4.2 Perform an Operation 139
  - 4.2(1) Simulate a Pervious Land Segment (Module PERLND) 142
    - 4.2(1).1 Correct Air Temperature for Elevation Difference (Section ATEMP) 142
    - 4.2(1).2 Simulate Accumulation and Melting of Snow and Ice (Section SNOW) 143
    - 4.2(1).3 Simulate Water Budget for a Pervious Land Segment (Section PWATER) 157
    - 4.2(1).4 Simulate Sediment Production and Removal (Section SEDMNT) 176
    - 4.2(1).5 Estimate Soil Temperatures (Section PSTEMP) 182
    - 4.2(1).6 Estimate Water Temperature and Dissolved Gas Concentrations (Section PWTGAS) . . . 133
    - 4.2(1).7 Simulate Quality Constituents Using Simple Relationships with Sediment and Water Yield (Section PQUAL) 184
    - 4.2(1).8 Estimate Moisture Content of Soil Layers and Fractional Fluxes of Solutes (Section MSTLAY) 192
    - 4.2(1).9 Simulate Pesticide Behavior in Detail (Section PEST) 195
    - 4.2(1).10 Simulate Nitrogen Behavior in Detail (Section NITR) 207
    - 4.2(1).11 Simulate Phosphorous Behavior in Detail (Section PHOS) 203
    - 4.2(1).12 Simulate Movement of a Tracer (Section TRACER) 205

4.2(2) Simulate an Impervious Land Segment (Module IMPLND) 207
  4.2(2).3 Simulate the Water Budget for an Impervious Land Segment (Section IWATER) . . . 207
  4.2(2).4 Simulate Accumulation and Removal of Solids (Section SOLIDS) 210
  4.2(2).5 Estimate Water Temperature and Dissolved Gas Concentrations (Section IWTGAS) . . . 214
  4.2(2).6 Simulate Washoff of Quality Constituents Using Simple Relationships with Solids and Water Yield (Section IQUAL) 215
4.2(3) Simulate a Free-flowing Reach or Mixed Reservoir (Module RCHRES) 219
  4.2(3).1 Simulate Hydraulic Behavior (Section HYDR) 219
  4.2(3).2 Prepare to Simulate Advection of Fully Entrained Constituents (Section ADCALC) . 238
  4.2(3).3 Simulate Conservative Constituents (Section CONS) 240
  4.2(3).4 Simulate Heat Exchange and Water Temperature (Section HTRCH) 244
  4.2(3).5 Simulate Inorganic Sediment (Section SED) 249
  4.2(3).6 Simulate First Order Decay Constituents (Section FSTORD) 254
  4.2(3).7 Simulate Constituents Involved in Biochemical Transformations 256
    4.2(3).7.1 Simulate Primary DO and BOD Balances (Section OXRX) . . . 258
    4.2(3).7.2 Simulate Primary Inorganic Nitrogen and Phosphorous Balances (Section NUTRX) . . 266
    4.2(3).7.3 Simulate Plankton Populations and Associated Reactions (Section PLANK) 273
    4.2(3).7.4 Simulate pH, Carbon Dioxide, Total Inorganic Carbon, and Alkalinity (Section PHCARB) . 296
4.2(11) Copy Time Series (Utility Module COPY) 302
4.2(12) Prepare Time Series for Display on a Plotter (Module PLTGEN) 302
4.2(13) Display Time Series in a Convenient Tabular Format (Utility Module DISPLY). 304
4.2(14) Perform Duration Analysis on a Time Series (Utility Module DURANL) 309
4.2(15) Generate a Time Series from One or Two Other Time Series (Utility Module GENER). 316
4.3 TSPUT Module. 317

References. 317

FIGURES

Number Page

3.0-1 Functions and data transfers involved in the operations portion of HSPF 136
4.2(1).2-1 Snow accumulation and melt processes 144
4.2(1).2-2 Flow diagram of water movement, storages and phase changes modeled in the SNOW section of the PERLND and IMPLND Application Modules 146
4.2(1).3-1 Hydrologic cycle 158
4.2(1).3-2 Flow diagram of water movement and storages modeled in the PWATER section of the PERLND Application Module . . . 160
4.2(1).3-3 Determination of infiltration and interflow inflow . . . 164
4.2(1).3-4 Fraction of the potential direct runoff retained by the upper zone (FRAC) as a function of the upper zone soil moisture ratio (UZRAT) 167
4.2(1).3-5 Fraction of infiltration plus percolation entering lower zone storage 172
4.2(1).3-6 Potential and actual evapotranspiration from the lower zone 175
4.2(1).4-1 Erosion processes 177
4.2(1).4-2 Flow diagram for SEDMNT section of PERLND Application Module 178
4.2(1).7-1 Flow diagram for PQUAL section of PERLND Application Module 186
4.2(1).8-1 Flow diagram of the transport of moisture and solutes, as estimated in the MSTLAY section of the PERLND Application Module 193
4.2(1).9-1 Flow diagram showing modeled movement of chemicals in solution 197
4.2(1).9-2 Freundlich isotherm calculations. 200
4.2(1).10-1 Flow diagram for nitrogen reactions 204
4.2(1).11-1 Flow diagram for phosphorus reactions 206
4.2(2)-1 Impervious land segment processes 208
4.2(2).3-1 Hydrologic processes 209
4.2(2).4-1 Flow diagram of the SOLIDS section of the IMPLND Application Module 212
4.2(2).6-1 Flow diagram for IQUAL section of IMPLND Application Module 216
4.2(3)-1 Flow of materials through a RCHRES 220
4.2(3).1-1 Flow diagram for the HYDR Section of the RCHRES Application Module 222
4.2(3).1-2 Graphical representation of the equations used to compute outflow rates and volume 225
4.2(3).1-3 Typical RCHRES configurations and the method used to represent geometric and hydraulic properties 226
4.2(3).1-4 Graphical representation of the work performed by subroutine ROUTE 229

4.2(3).1-5 Graphical representation of the work performed by subroutine NOROUT 234
4.2(3).1-6 Illustration of quantities involved in calculation of depth 236
4.2(3).2-1 Determination of weighting factors for advection calculations. 240
4.2(3).3-1 Flow diagram for conservative constituents in the CONS section of the RCHRES Application Module. 240
4.2(3).4-1 Flow diagram for HTRCH section of the RCHRES Application Module 245
4.2(3).5-1 Flow diagram for washload in the SED section of the RCHRES Application Module 250
4.2(3).5-2 Flow diagram for sandload in the SED section of the RCHRES Application Module 251
4.2(3).6-1 Flow diagram for first order decay constituents in the FSTORD section of the RCHRES Application Module 255
4.2(3).7.1-1 Flow diagram for dissolved oxygen in the OXRX subroutine group of the RCHRES Application Module 259
4.2(3).7.1-2 Flow diagram for biochemical oxygen demand in the OXRX subroutine group of the RCHRES Application Module 260
4.2(3).7.2-1 Flow diagram for inorganic nitrogen in the NUTRX subroutine group of the RCHRES Application Module 267
4.2(3).7.2-2 Flow diagram for ortho-phosphate in the NUTRX group of the RCHRES Application Module 268
4.2(3).7.3-1 Flow diagram for phytoplankton in the PLANK section of the RCHRES Application Module 274
4.2(3).7.3-2 Flow diagram for dead refractory organics in the PLANK section of the RCHRES Application Module 274
4.2(3).7.3-3 Flow diagram for zooplankton in the PLANK section of the RCHRES Application Module 275
4.2(3).7.3-4 Flow diagram for benthic algae in the PLANK section of the RCHRES Application Module 275
4.2(3).7.3-5 Relationship of parameters for special advection of plankton 277
4.2(3).7.3-6 Zooplankton assimilation efficiency 259
4.2(3).7.4-1 Flow diagram of inorganic carbon in the PHCARB section of the RCHRES Application Module 297
4.2(13)-1 Sample Short-Span Display (First Type). 306
4.2(13)-2 Sample Short-Span Display (Second Type) 307
4.2(13)-3 Sample Long-Span (Annual) Display 308
4.2(14)-1 Definition of Terms Used in Duration Analysis Module. . . 310
4.2(14)-2 Sample Duration Analysis Printout 312

GENERAL COMMENTS

For a discussion on how this part of the documentation is organized, refer to Section 5 in Part C "Standards and Conventions."

The subprograms are discussed in numerical order, following the numbering system used on the structure charts (Part D). When studying the descriptions which follow, you will find it helpful to refer to the appropriate structure chart and to the pseudo code in the Programmer's Supplement (Johanson, et al. 1979).

1.0 MAIN Program

The MAIN program stands at the head of the structure (Structure Chart 1.0) and calls, directly or indirectly, all the other modules in the system. The functions performed are:

(1) Preprocess the Users Control Input (UCI). Subroutine USRRDR transfers the UCI to a direct access file, appends a value at the end of each record which points to the next non-comment record, and recognizes data set headings and delimiters: RUN, END RUN, TSSM, END TSSM.

(2) Call TSSMGR if a TSSM data set has been found.

(3) If a RUN data set has been found, call INTERP to interpret it and then call OSUPER to supervise execution of it.

2.0 Manage the Time Series Store (Module TSSMGR)

General Description of Module TSSMGR

This module maintains a user's Time Series Store (TSS) and performs some housekeeping chores associated with the data sets in it. From the point of view of the computer's operating system, the TSS is a single file (which may be very large). However, the HSPF software can store many distinct time series in this file. This permits a user easily to keep track of the various time series with which he is dealing. Furthermore, he need only refer to a single disc file for all his time series input and output needs, no matter how many time series are involved. This simplifies communication with the computer system.

Time series are arranged within the TSS in one or more "data sets". The contents of each data set and its physical location in the TSS are recorded in a directory located at the start of the TSS. The primary function of module TSSMGR is to maintain this directory, from input supplied by the user. He can add a new data set label to the directory, update certain parts of the label (such as the SECURITY option), scratch a data set label (and, thus, the data set contents), extend the space allocated to a data set, or show the contents of one or all of the labels in the directory. The commands used to achieve this are documented in Part F, Section 2.

The design of the TSS is based on our experience with HSPX and HSPII. Extensions to the HSPX time series management system include:

(1) The storage of data in compressed form. Disk space is saved by improving the method of encoding values which occur in a sequence, such as a string of zeros.

(2) The inclusion of multimembered data sets in the TSS. This permits storage of several time series in the same data set, which is useful if one needs to save several related functions of time. For example, if a user needs to save the discharge of water and associated constituents from a stream reach, he needs to create and refer to only one data set (with several members). Up to 20 time series may be stored in a single data set.

Further details of the organization of a TSS are given in the documentation for program NEWTSS (Appendix III).

Once the label for a dataset has been created and space reserved for it in the TSS, time series data can be stored in the data set. This is done by an operating module (Section 4.0); either a utility module (eg. COPY) or an application module (eg. PERLND).

3.0 Interpret a RUN Data Set in the User's Control Input (Module INTERP)

General Description of Module INTERP

This module, known as the Run Interpreter, translates a RUN data set in the User's Control Input (documented in Section 4 of Part F) into many elementary instructions, for later use by other parts of the system, when the time series are operated on. To do this, the Run Interpreter performs such tasks as:

(1) Check and augment the data supplied by the user.

(2) Decide which time series will be required and produced by each operation, based on the user's data and built-in tables which contain information on the various operations.

(3) Allocate INPAD rows to the various time series.

(4) Read the control data, parameters, and initial conditions supplied for each opertion, convert them to internal units, and supply default values where required.

The output of the Run Interpreter is stored in disk-based files containing instructions to be read by the Operations Supervisor, TSGET and TSPUT (Figure 3.0-1). The instruction files are:

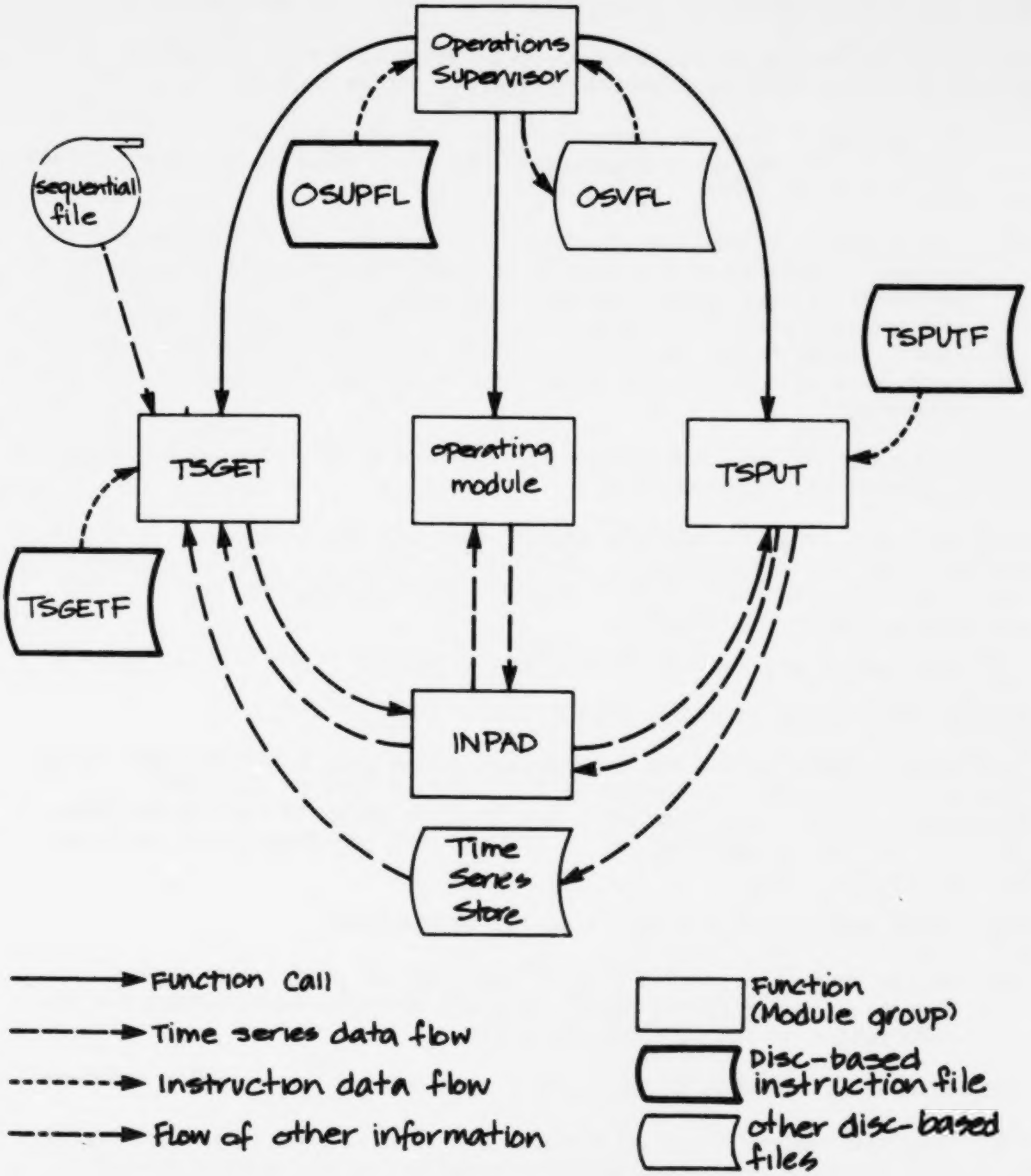


FIG. 3.0-1 Functions and data transfers involved in the operations portion of HSPF

(1) The Operations Supervisor Instruction File (OSUPFL). This file contains instructions which the Operations Supervisor reads to manage the operations in a run. This includes information on:

  (a) the configuration of the scratch pads (time intervals and widths)

  (b) the configuration of the EXGROUPs and INGROUPs (number of EXGROUPs, number INGROUPs in each EXGROUP, operations in each INGROUP, etc.) (EXGROUPs have not yet been implemented)

(2) The Operation Status Vector File (OSVFL). The operations in a run are interrupted every time an INPAD span is completed (Part B, Section 3.2). To save machine core space, the system is designed so that the various operations all use the same area of core. This requires that upon interruption, all information necessary to restart an operation be stored in a disk file. The data, called the "Operation Status Vector" (OSV), reside in a string of contiguous locations in core and have a structure specified in the Programmer's Supplement (Johanson, et al. 1979). The disc file OSVFL contains an exact copy of the OSV for each operation. It is used to restore the OSV in core when the operation is resumed after interruption.

(3) The Input Time Series Instruction File (TSGETF) and the Output Time Series Instruction File (TSPUTF). These files contain instructions which govern the transfer of pieces of time series into and out of the INPAD, respectively. Each instruction enables module TSGET to retrieve a specified piece of time series from one of the source volumes (Figure 3.0-1), transform it to the interval and form required for the INPAD, and insert it in the desired row of the INPAD. In the case of TSPUTF, the sequence is the reverse of that just described.

    Each operation has its own set of instructions in TSGETF and TSPUTF which are read whenever modules TSGET and TSPUT are called upon to service that operation (every INSPAN).

(4) The Special Action Instruction File (SPACFL). Each record of this file contains a single special action instruction, which specifies the action required to be taken in a given operation at a specific time, e.g. report operation state, modify a state variable. (not yet implemented)

The structures of these files are documented in the Programmer's Supplement.

4.0 Supervise and Perform Operations (module OSUPER)

Function of Operations Group

The Operations group of modules handles all the manipulations of time series and thus, performs most of the work in a run. Subroutine OSUPER controls the group. It performs some of the tasks itself, but it invokes subordinate modules to do other tasks.

General Description of Subroutine OSUPER

The primary tasks of subroutine OSUPER are to ensure that the various operations in the run are called in the correct sequence and that the associated time series and OSVs are input and/or output at the required junctures (see Part B, Section 3.2). OSUPER uses a nest of DO-loops to control the sequencing. The instruction file OSUPFL specifies the ranges of the loops and supplies information ("keys") which enable OSUPER, TSGET and TSPUT to correctly access the other instruction files. OSUPER reads an instruction from disc each time an operation starts a new INSPAN. Using this information, it then:

(1) calls TSGET, to supply the required input time series (using TSGKST, TSGKND)

(2) reads the OSV from disc (using keys OSVKST, OSVKND)

(3) calls the operating module (OMCODE indicates which one is to be called)

When the INSPAN is over, OSUPER:

(1) writes the OSV to disc (using keys OSVKST, OSVKND)

(2) calls TSPUT, to output time series (using keys TSPKST, TSPKND)

4.03 Perform Special Actions (Subroutine SPECL)

HSPF permits the user to perform certain "Special Actions" during the course of a run. A special action instruction specifies the following:

1. The operation on which the action is to be performed (eg. PERLND 10)
2. The date/time at which the action is to be taken.
3. The type and location, within COMMON block SCRTCH, of the data item to be updated.
4. The action to be performed. Two choices are available:
    a) Reset the variable to a specified value
    b) Increment the variable by a specified value

The special action facility is used to accomodate things such as:

1. Human intervention in a watershed. Events such as plowing, cultivation, fertilizer and pesticide application, and harvesting are simulated in this way.
2. Changes to parameters. For example, a user may wish to alter the value of a parameter for which 12 monthly values cannot be supplied. He can do this by specifying a special action for that variable. He could reset the parameter to its original value by specifying another special action, to be taken at a later time.

At present, Special Actions can only be performed on variables in the PERLND module. The input is documented in Section 4.10 of Part F.

4.1 Get Required Input Time Series (module TSGET)

The task of this module is to insert in the INPAD all input time series required by an operation. OSUPER calls it (once) each time an operation is to commence an INSPAN, passing to it the keys of the first and last records in TSGETF which must be read and acted upon. Each instruction causes a row of the INPAD to be filled. TSGET can draw its input time series from any of the following source "volumes": TSS, sequential file and INPAD (Figure 3.0-1).

TSGET will, if necessary, automatically transform the time interval and "kind" (Appendix III) of the time series, as it is brought from the source location to the INPAD (target). TSGET can also perform a linear transformation on the values in a time series; for example, if the source contains temperatures in degrees C and the INPAD needs them in degrees F.

4.2 Perform an Operation

Function of an Operating Module

An operating module (OM) is at the center of every operation (Part B, Section 2.1). When the Operations Supervisor calls an OM the time series which it requires are already in the INPAD. The task of the OM is to operate on these input time series. The results of this work are:

(1) updated state variables. The OM constantly updates any state variables. These are located in the OSV. Thus, when the OM returns control to the Operations Supervisor, which copies the OSV to disc, the latest values of all state variables are automatically preserved.

(2) printed output. The OM accumulates values, formats them and routes these data to the line printer.

(3) output time series. The OM places these in the INPAD but is not concerned with their ultimate disposition; this is handled by module TSPUT.

Note that all time series simultaneously present in an INPAD have the same constant time interval. This implies that, internally, all time series involved in an operation have the same time interval. Externally, the time series may have differing time intervals. Part of the function of modules TSGET and TSPUT is to convert time series from external to internal time intervals and vice versa.

Sub-divisions in an Operating Module

An operating module may be divided into several distinct "sections," each of which may be selectively activated in a given run, under the user's control, e.g. the Pervious Land-segment module (PERLND) contains twelve sections, the first being air temperature correction, the last tracer (conservative substance) simulation (Structure Chart 4.2(1)). The operating procedure is as follows: in each time interval of the INSPAN, the operating module calls each of its active sections in the order in which they are built into the code (the sequence can not be altered by the user). When the INSPAN has been covered the operating module returns control to OSUPER which determines the next action to be taken. This procedure implies that an operating module must be arranged so that a section is called after any others from which it requires information. For example, in the Pervious Land-segment module (Structure Chart 4.2(1)), the sediment calculation section may use data computed by the snow and water balance sections but not by Sections 5 through 10. This kind of information flow is called an inter-section data transfer (ISDT).

Partitioning of an Operation

A user may activate one group of module sections in an initial run and other groups in subsequent runs. Thus, he may "partition" an operation. For example, he may wish to calibrate the hydraulic response of a set of river reaches before moving on to simulate the behavior of constituents contained in the water. If this type of work involves ISDT's between the sections handled in different runs, it follows that:

(1) The time series involved in the ISDT's must be stored between runs, probably in the TSS.

(2) In the second run the system will expect the user to specify external sources for all these time series.

Some users will be confused by the rules for partitioning operations, but our experience indicates this will be outweighed by the flexibility which it brings.

Numbering of Operating Modules

In principle, there is no limit to the number of operating modules which the system can accommodate. Ultimately, we expect a large number of modules ranging from very simple utility modules (eg. COPY) to very complex simulation algorithms (eg. PERLND). Although the size and complexity of the modules vary greatly, they all are, logically, of equal rank (Figure 2-3, Part B). The adopted numbering system reflects this. Every operating module is identified by the number 4.2 (Structure Chart 4.0) and is distinguished from the others only by a subscript. For example, the Pervious Land-segment Module is 4.2(1) and the Reach/Mixed Reservoir Module 4.2(3).

Inserting Additional Operating Modules

A user may insert additional modules. To do this he must:

(1) Write or adapt his operating module. This includes restructuring the data into an OSV which conforms with the requirements of the HSPF system. (This task may be time consuming).

(2) Add a section of code to the Run Interpreter to interpret the UCI for the new module.

(3) Add data to the information file (INFOFL) and, if necessary, to the warning and error message files.

(4) Make minor changes to subroutines OPNBLK and OSUPER.

Types of Operating Modules

There are two types of operating modules; utility modules and application modules. Utility modules perform any operations involving time series which are essentially auxiliary to application operations, e.g. input time series data from cards to the TSS using COPY, multiply two time series together to obtain a third one, plot several time series on the same graph. The utility modules perform many of the functions which were previously part of HSP LIBRARY or HSP UTILITY. They are given numbers starting with 4.2(11). Application (simulation) modules represent processes, or groups of processes, which occur in the real world. They have been allocated numbers 4.2(1) through 4.2(10) although, at present, only three application modules are written.

4.2(1) Simulate a Pervious Land Segment (Module PERLND)

A land segment is a subdivision of the simulated watershed. The boundaries are established according to the user's needs, but generally, a segment is defined as an area with similar hydrologic characteristics. For modeling purposes water, sediment, and water quality constituents leaving the watershed move laterally to a downslope segment or to a reach/reservoir. A segment of land which has the capacity to allow enough infiltration to influence the water budget is considered pervious. In HSPF, PERLND is the module that simulates the water quality and quantity processes which occur on a pervious land segment.

The primary module sections in PERLND simulate snow accumulation and melt (Section SNOW), the water budget (section PWATER), sediment produced by land surface erosion (section SEDMNT), and water quality constituents by various methods (section PQUAL and the agri-chemical sections). Other sections perform the auxiliary functions of correcting air temperature (section ATEMP) for use in snowmelt and soil temperature calculations, producing soil temperatures (section PSTEMP) for estimating the outflow temperatures and influencing reaction rates in the agri-chemical sections, and determining outflow temperatures which influence the solubility of oxygen and carbon dioxide. Structure Chart 4.2(1) shows these sections and their relationships to each other and to PPTOT, PBAROT, and PPRINT. The last three sections manipulate the data produced. Section PPTOT places state varables (point values) and PBAROT places flux variables which are actually averages over the interval (bar values) into the INPAD. PPRINT produces the printable results in the quantity and frequency that the use specifies. The sections in the structure chart are executed from left to right.

4.2(1).1 Correct Air Temperature for Elevation Difference (Section ATEMP of Modules PERLND and IMPLND)

Purpose

The purpose of ATEMP is to modify the input air temperature to represent the mean air temperature over the land segment. This module section is part of both PERLND or IMPLND. Air temperature correction is needed when the elevation of the land segment is significantly different than the elevation at the temperature gage. If no correction for elevation is needed, this module section can be skipped.

Method

The lapse rate for air temperature is dependent upon precipitation during the time interval. If precipitation occurs, a wet lapse rate of 0.0035 degrees F per foot difference in elevation is assumed. Otherwise, a dry lapse rate, that varies with the time of day, is used. A table of 24 hourly dry lapse rates varying between 0.0035 to 0.005 is built into the system. The corrected air temperature is:

$$AIRTMP = GATMP - LAPS*ELDAT \quad (1)$$

where:

AIRTMP = corrected air temperature in degrees F
GATMP = air temperature at gage in degrees F
LAPS = lapse rate in degrees F/ft
ELDAT = elevation difference between the land segment and the gage in ft

4.2(1).2 Simulate Accumulation and Melting of Snow and Ice (section SNOW of modules PERLND and IMPLND)

Purpose

SNOW deals with the runoff derived from the fall, accumulation and melt of snow. This is a necessary part of any complete hydrologic package since much of the runoff, especially in the northern half of the United States, is derived from snow conditions.

Approach

Figure 4.2(1).2-1 illustrates the processes involved in snow accumulation and melt on a land segment. The algorithms used are based on the work by the Corps of Engineers (1956), Anderson and Crawford (1964), and Anderson (1968). Empirical relationships are employed when physical ones are not well known. The snow algorithms use meteorologic data to determine whether precipitation is rain or snow, to simulate an energy balance for the snowpack, and to determine the effect of the heat fluxes on the snowpack.

Five meteorologic time series are required by SNOW for each land segment simulated. They are:

precipitation
air temperature
solar radiation
dewpoint
wind velocity

A value from each of these time series is input to SNOW at the start of each simulation interval. However, some of the meteorological time series are only used intermittently for calculating rates, such as in the calculation of the potential rate of evaporation from the snowpack.

Air temperature is used to determine when snow is falling. Once snow begins to accumulate on the ground, the snowpack accumulation and melt calculations take place. Five sources of heat which influence the melting of the snowpack are simulated:

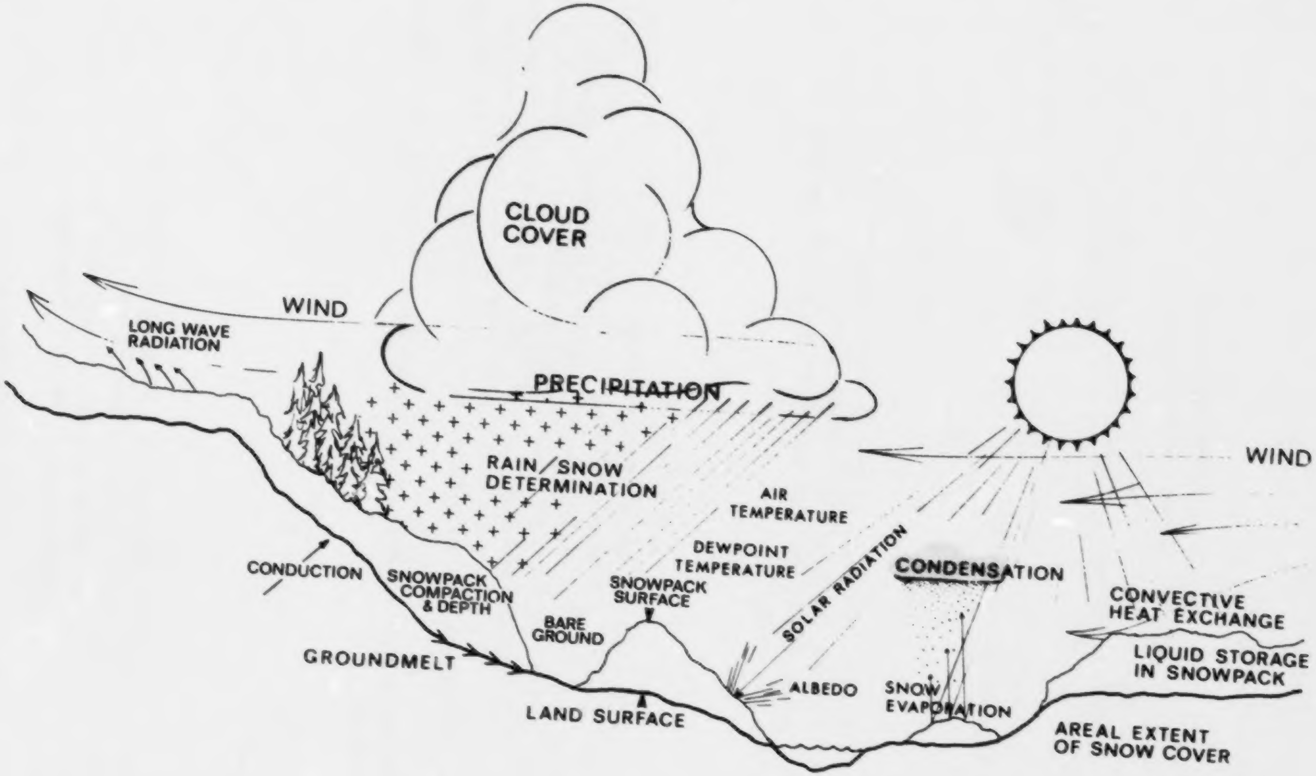


Figure 4.2(1).2-1 Snow accumulation and melt processes

1. net radiation heat (RADHT), both longwave and shortwave
2. convection of sensible heat from the air (CONVHT)
3. latent heat transfer by condensation of moist air on the snowpack (CONDHT)
4. heat from rain, sensible heat from rain falling (RNSHT) and latent heat from rain freezing on the snowpack
5. conduction of heat from the underlying ground to the snowpack (GMELTR)

Other heat exchange processes such as latent heat from evaporation are considered less significant and are not simulated.

The energy calculations for RADHT, CONVHT, and CONDHT are performed by subroutine HEXCHR while GMELTR is calculated in subroutine GMELT. Latent heat from rain freezing is considered in subroutine WARMUP. RNSHT is computed in the parent subroutine SNOW. For uniformity and accounting, energy values are calculated in terms of the water equivalent which they could melt. It takes 202.4 calories per square cm on the surface to melt one inch water equivalent of snow at 32 degrees F. All the sources of heat including RNSHT are considered to be positive (incoming to the pack) or zero, except RADHT which can also be negative (leaving the pack).

Net incoming heat from the atmosphere (the sum of RADHT, CONVHT, CONDHT, and RNSHT) is used to warm the snowpack. The snowpack can be further warmed by the latent heat released upon rain freezing. Any excess heat above that required to warm the snowpack to 32 degrees F is used to melt the pack. Likewise, net loss of heat is used to cool the snowpack producing a negative heat storage. Futhermore, incoming heat from the ground melts the snowpack from the bottom independent of the atmospheric heat sources except that the rate depends on the temperature of the snowpack.

Figure 4.2(1).2.2 gives a schematic view of the moisture related processes modeled in section SNOW. Precipitation may fall as rain or snow on the snowpack or the ground. Evaporation only occurs from the frozen portion of the pack (PACKF). The frozen portion of the pack is composed of snow and ice. The ice portion of PACKF is considered to be in the lower part of the snowpack, so it is the first to melt when heat is conducted from the ground. Similarly, the snow portion of PACKF is the first to melt when atmospheric heat increases. Melted PACKF and rain falling on the snowpack produce the water portion of the total snowpack which may overflow the capacity of the pack. The water yield and rain on the bare ground becomes input to module section PWATER or IWATER. These moisture related processes as well as the heat exchange processes are discussed later in more detail.

Heat transfer from incoming rain (RNSHT) to the snowpack is calculated in the parent subroutine SNOW (Section 4.2(1).2). The following physically based equation is used:

```
RNSHT = (AIRTMP - 32.0)*RAINF/144.0                          (2)
```

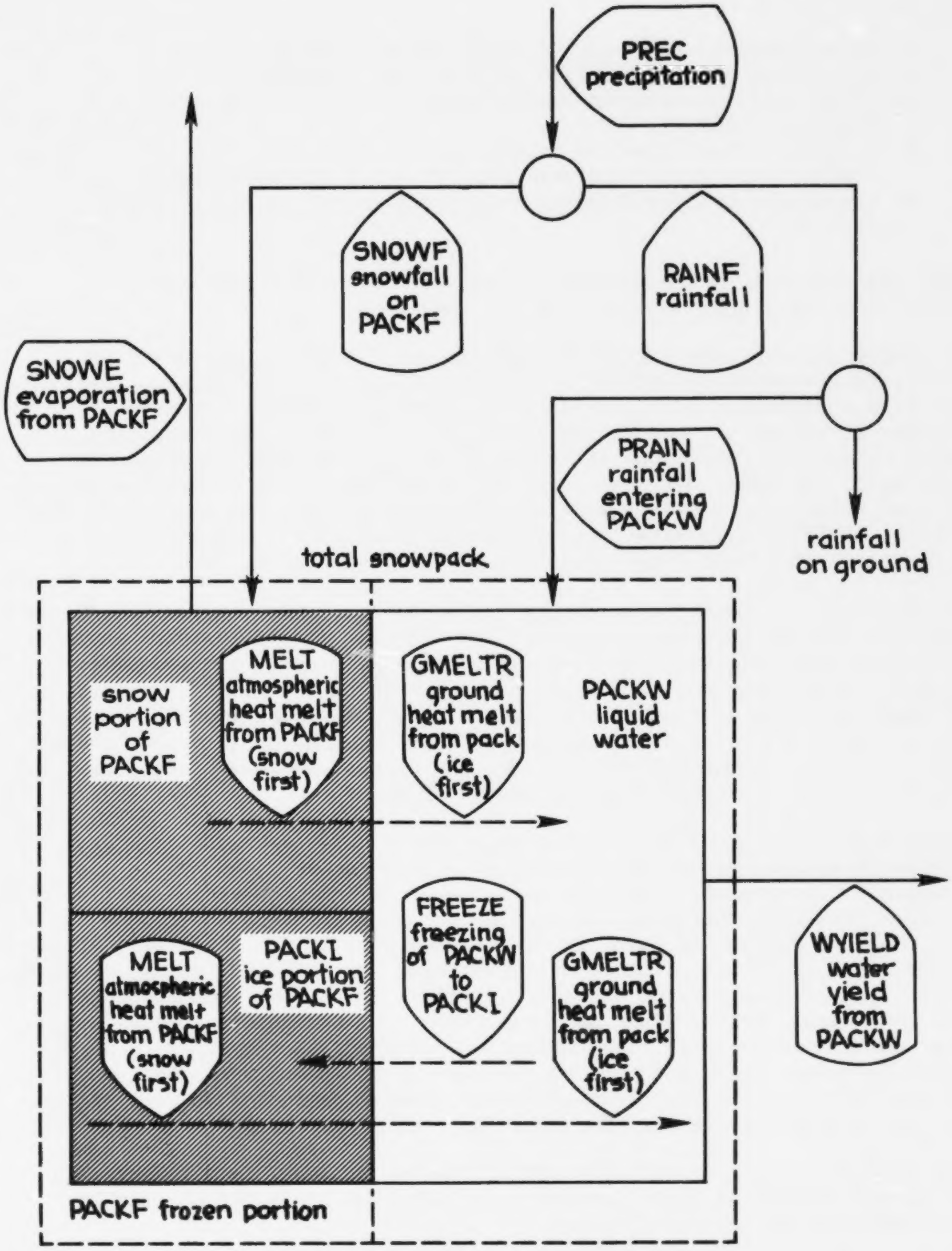


Figure 4.2(1).2-2 Flow diagram of water movement, storages and phase changes modeled in the SNOW section of the PERLND and IMPLND Application Modules

where:

AIRTMP = temperature of the air in degrees F
RAINF = rainfall in inches
144.0 = factor to convert to equivalent depth of melt
32.0 = freezing point in degrees F

Other characteristics of the snowpack are also determined in the main subroutine SNOW. The fraction of the land segment covered by the snowpack is estimated by merely dividing the depth of the snowpack by a cover index (COVINX) which is a function of the parameter COVIND and the history of the pack as explained in subroutine EFFPRC. The temperature of the snowpack is estimated by:

$$PAKTMP = 32.0 - NEGHTS/(0.00695*PACKF) \qquad (3)$$

where:

PAKTMP = mean temperature of the snowpack in degrees F
NEGHTS = negative heat storage in inches of water equivalent
PACKF = frozen contents of the snowpack in inches of water equivalent
0.00695 = physically based conversion factor

4.2(1).2.1 Estimate Meteorological Conditions (subroutine METEOR)

Purpose

Subroutine METEOR estimates the effects of certain meteorological conditions on specific snow-related processes by the use of empirical equations. It determines whether precipitation is falling as snow or rain. The form of precipitation is critical to the reliable simulation of runoff and snowmelt. When snow is falling, the density is calculated in order to estimate the depth of the new snowpack. The fraction of the sky which is clear is also estimated for use in the radiation algorithms, and the gage dewpoint is corrected if it is warmer than air temperature.

Method

The following expression is used to calculate hourly the effective air temperature below which snowfall occurs:

$$SNOTMP = TSNOW + (AIRTMP - DEWTMP)*(0.12 + 0.008*AIRTMP) \qquad (4)$$

where:

SNOTMP = air temperature below which snowfall occurs in degrees F
TSNOW = parameter in degrees F
AIRTMP = air temperature in degrees F
DEWTMP = dewpoint in degrees F

SNOTMP is allowed to vary in this calculation by a maximum of one degree F from TSNOW. When AIRTMP is equal to or greater than SNOTMP, precipitation is assumed to be rain.

When snowfall occurs, its density is estimated as a function of air temperature according to:

RDNSN = RDCSN + (AIRTMP/100.0)**2 (5)

where:
RDNSN = density of new snowfall (at zero degrees F or greater) relative to liquid water
RDCSN = parameter designating density of new snow at an air temperature of zero degrees F and lower, relative to liquid water

RDNSN is used in subroutine EFFPRC to calculate the new depth of the snowpack resulting from the addition of the snow. This and all other snow density terms are in water equivalent (inches) per depth of the snowpack (inches).

The fraction of the sky which is clear (SKYCLR) is needed for the calculation of the longwave back radiation to the snowpack from the clouds (done in subroutine HEXCHR). SKYCLR is set to the minimum value of 0.15 when precipitation occurs. Otherwise, it is increased each simulation time interval as follows:

SKYCLR = SKYCLR + (0.0004*DELT) (6)

where:
DELT = simulation time interval in min

SKYCLR increases until either it reaches unity or precipitation causes it to be reset.

A gage dewpoint higher than air temperature is not physically possible and will give erroneous results in the calculation of snowpack evaporation. Therefore, dewpoint is set equal to the air temperature when this situation occurs. Otherwise, the gage dewpoint is used.

4.2(1).2.2 Determine the Effect of Precipitation on the Pack (subroutine EFFPRC)

Purpose

The purpose of this subroutine is to add the falling snow to the pack, determine the amount of rain falling on the snowpack, and adjust the snowpack dullness to take into account new snow.

Method

The amount of precipitation falling as snow or rain is determined in subroutine METEOR. Subroutine EFFPRC accounts for the influence that snowfall and rain have on the land segment. The subroutine begins by increasing the snowpack depth by the amount of snow falling on the pack divided by its density.

The fraction of the land segment which is a covered by the snowpack (SNOCOV) is determined by re-evaluating the index to areal coverage (COVINX). When the frozen contents of the pack (PACKF) exceeds the value of the parameter describing the maximum PACKF required to insure complete areal coverage by snow cover (COVIND), then COVINX is set equal to COVIND. Otherwise, COVINX is equal to the largest previous value of PACKF. SNOCOV is PACKF/COVINX if PACKF < COVINX. The amount of rain falling on the snowpack is that fraction of the precipitation which falls as rain multiplied by the SNOCOV. Rain falling on the snowpack will either freeze, adding to the frozen portion of the pack and produce heat used to warm the pack (see subroutine WARMUP), or it will increase the liquid water content of the pack (see subroutine LIQUID). Any rain not falling on the pack is assumed to land on bare ground.

When snowfall occurs, the index to the dullness of the snowpack (DULL) is decreased by one thousand times the snowfall for that interval. However, if one thousand times the snowfall is greater than the previous value for DULL, then DU LL is set to zero to account for a new layer of perfectly reflectable snow. Otherwise, when snowfall does not occur, DULL is increased by one index unit per hour up to a maximum of 800. Since DULL is an empirical term used as an index, it has no physical units. DULL is used to determine the albedo of the snowpack which in turn is used in the shortwave energy calculations in subroutine HEXCHR.

4.2(1).2.3 Compact the Pack (subroutine COMPAC)

Purpose

The addition of new snow will reduce the density as well as increase the depth of the snowpack as in subroutine EFFPRC. The pack will tend to compact with age until a maximum density is reached. The purpose of subroutine COMPAC is to determine the rate of compaction and calculate the actual change in the depth due to compaction.

Method

When the relative density is less than 55 percent compaction is assumed to occur. The rate of compaction is computed according to the empirical expression:

$$\text{COMPCT} = 1.0 - (0.00002*\text{DELT60}*\text{PDEPTH}*(0.55 - \text{RDENPF})) \tag{7}$$

where:

COMPCT = unit rate of compaction of the snowpack per interval
DELT60 = number of hours in an interval
PDEPTH = depth of the snowpack in inches of total snowpack
RDENPF = density of the pack relative to liquid water

The new value for PDEPTH is COMPCT times PDEPTH. PDEPTH is used to calculate the relative density of the snowpack which affects the liquid water holding capacity as determined in subroutine LIQUID.

4.2(1).2.4 Simulate Evaporation from the Pack (subroutine SNOWEV)

Purpose

The SNOWEV subroutine estimates evaporation from the snowpack (sublimation).

Method

Evaporation from the snowpack will occur only when the vapor pressure of the air is less than that of the snow surface, that is, only when the air vapor pressure is less than 6.108 mbar which is the maximum vapor pressure that the thin surface film of air over the snowpack can attain. When this condition is met the evaporation is computed by the empirical relationship:

$$SNOWEP = SNOEVP*0.0002*WINMOV*(SATVAP - VAP)*SNOCOV \qquad (8)$$

where:

SNOWEP = potential rate of evaporation from the frozen part of the snowpack in inches of water equivalent/interval
SNOEVP = parameter used to adjust the calculation to field conditions
WINMOV = wind movement in miles/interval
SATVAP = saturated vapor pressure of the air at the current air temperature in mbar
VAP = vapor pressure of the air at the current air temp, in mbar
SNOCOV = fraction of the land segment covered by the snowpack

The potential (SNOWEP) will be fulfilled if there is sufficient snowpack. Otherwise, only the remaining pack will evaporate. For either case, evaporation occurs only from the frozen content of the snowpack (PACKF).

4.2(1).2.5 Estimate Heat Exchange Rates (except ground melt and rain heat) (subroutine HEXCHR)

Purpose

The purpose of this subroutine is to estimate the heat exchange from the atmosphere due to condensation, convection, and radiation. All heat exchanges are calculated in terms of equivalent depth of melted or frozen water.

Method of Determining Heat Supplied by Condensation

Transfer of latent heat of condensation can be important when warm moist air masses travel over the snowpack. Condensation occurs when the air is moist enough to condense on the snowpack. That is, when the vapor pressure of the air is greater than 6.108 mbar. This physical process is the opposite of snow evaporation; the heat produced by it is calculated by another empirical relationship:

```
     CONDHT = 8.59*(VAP - 6.108)*CCFACT*0.00026*WINMOV              (9)
where:
     CONDHT = condensation heat flux to the snowpack in inches of water
              equivalent/interval
     VAP    = vapor pressure of the air at the current air temp, in mbar
     CCFACT = parameter used to correct melt values to field conditions
     WINMOV = wind movement in miles/interval
```

CONDHT can only be positive or zero, that is, incoming to the pack.

Method of Determining Heat Supplied by Convection

Heat supplied by turbulent exchange with the atmosphere can occur only when air temperatures are greater than freezing. This convection of heat is calculated by the empirical expression:

```
     CONVHT = (AIRTMP - 32.0)*(1.0 - 0.3*MELEV/10000.0)*            (10)
              CCFACT*0.00026*WINMOV
where:
     CONVHT = convective heat flux to the snowpack in inches of water
              equivalent/interval
     AIRTMP = air temperature in degrees F
     MELEV  = mean elevation of the land segment above sea level in ft
```

In the simulation, CONVHT also can only be positive or zero, that is, only incoming.

Method of Determining Heat Supplied by Radiation

Heat supplied by radiation is determined by:

```
     RADHT = (SHORT + LONG)/203.2                                   (11)
where:
     RADHT = radiation heat flux to the snowpack in inches of
             water equivalent/interval
     SHORT = net solar or shortwave radiation in langleys/interval
     LONG  = net terrestrial or longwave radiation in langleys/interval
```

The constant 203.2 is the number of langleys required to produce one inch of melt from snow at 32 degrees F. RADHT can be either positive or negative, that is, incoming or outgoing.

SHORT and LONG are calculated as follows. Solar radiation, a required time series, is modified by the albedo and the effect of shading. The albedo or reflectivity of the snowpack is a function of the dullness of the pack (see subroutine EFFPRC for a discussion of DULL) and the season. The equation for calculating albedo (ALBEDO) for the 6 summer months is:

ALBEDO = 0.80 - 0.10*(DULL/24.0)**0.5 (12)

The corresponding equation for the winter months is:

ALBEDO = 0.85 - 0.07*(DULL/24.0)**0.5 (13)

ALBEDO is allowed a minimum value of 0.45 for summer and 0.60 for winter. The hemispheric location of the land segment is taken into account for determining summer and winter in using the above equation. This is done through the use of the latitude parameter which is positive for the northern hemisphere.

Once the albedo of the pack is found then solar radiation (SHORT) is modified according to the equation:

SHORT = SOLRAD*(1.0 - ALBEDO)*(1.0 - SHADE) (14)

where:

SOLRAD = solar radiation in langleys/interval
SHADE = parameter indicating the fraction of the land segment which is shaded

Unlike shortwave radiation which is more commonly measured, longwave radiation (LONG) is estimated from theoretical consideration of the emitting properties of the snowpack and its environment. The following equations are based on Stefan's law of black body radiation and are linear approximations of curves in Plate 5-3, Figure 6 in Snow Hydrology (Corps of Engineers 1956). They vary only by the constants which depend on air temperature. For air temperatures above freezing:

LONG = SHADE*0.26*RELTMP + (1.0 - SHADE)*(0.2*RELTMP - 6.6) (15)

And for air temperatures at freezing and below:

LONG = SHADE*0.20*RELTMP + (1.0 - SHADE)*(0.17*RELTMP - 6.6) (16)

where:

RELTMP = air temperature minus 32 in degrees F
6.6 = average back radiation lost from the snowpack in open areas in langleys/hr

Since the constants in these equations were originally based on hourly time steps, both calculated values are multiplied by DELT00, the number of hours per interval, so that they correspond to the simulation interval. In addition, LONG is multiplied by the fraction of clear sky (SKYCLR) when it is negative to account for back radiation from clouds.

4.2(1).2.6 Simulate Loss of Heat from Pack (subroutine COOLER)

Purpose

The purpose of this code is to cool the snowpack whenever it is warmer than the ambient air and thus loses heat. This is accomplished by accumulating negative heat storage which increases the capacity of the pack to later absorb heat without melting as simulated in subroutine WARMUP.

Method

In every interval where there is heat loss to the atmosphere and the temperature of the snowpack is greater than the air temperature, the negative heat storage will increase; that is, the pack will cool. However, there is a maximum negative heat storage. The maximum negative heat storage that can exist at any time is found by assuming a linear temperature distribution from the air temperature which is considered to be above the pack to 32 degrees at the bottom of the snowpack. This maximum negative heat storage is calculated hourly as follows:

MNEGHS = 0.00695*(PACKF/2.0)*(-RELTMP) (17)

where:

MNEGHS = maximum negative heat storage, inches of water equivalent
PACKF = water equivalent of the frozen contents of the snowpack, inches
RELTMP = air temperature above freezing in degrees F

The accummulation of the negative heat storage is calculated hourly from the following empirical relationship:

NEGHT = 0.0007*(PAKTMP - AIRTMP)*DELT60 (18)

where:

NEGHT = potential rate of cooling of the snowpack in inches of water equivalent per interval
PAKTMP = mean temperature of the snowpack in degrees F
AIRTMP = air temperature in degrees F
DELT60 = number of hours per interval

NEGHT is added to the negative heat storage (NEGHTS) every interval except when limited by MNEGHS. NEGHTS is used in the parent subroutine SNOW to calculate the temperature of the snowpack and in subroutine WARMUP to determine the extent that the pack must be warmed to reach 32 degrees F.

4.2(1).2.7 Warm the Snowpack if Possible (subroutine WARMUP)

Purpose

This subroutine warms the snowpack to as much as 32 degrees F when possible.

Method

When there is negative heat storage in the pack (see subroutine COOLER for a discussion of NEGHTS), and there is net incoming energy as calculated in previous subroutines, then NEGHTS will decrease resulting in a warmer snowpack and possible melt.

The calculations in this subroutine are merely accounting. They decrease NEGHTS to a minimum of zero by subtracting the net incoming heat. If any negative heat storage remains, then the latent heat released by the freezing of any incoming rain is added to the pack. Since NEGHTS and all other heat variables are in units of inches of melt, the inches of rain falling on the pack and freezing is subtracted from NEGHTS without any conversion.

4.2(1).2.8 Melt the Pack Using Any Remaining Heat (subroutine MELTER)

Purpose

MELTER simulates the actual melting of the pack with whatever incoming heat remains. Any heat which was not used to heat the snowpack in subroutine WARMUP can now be used to melt the snowpack.

Method

This subroutine is also merely an accounting subroutine. The net incoming heat has already been calculated in terms of water equivalents of melt. Hence, any remaining incoming heat is used directly to melt the snowpack either partially or entirely depending on the size of the snowpack.

4.2(1).2.9 Handle Liquid Water in the Pack (subroutine LIQUID)

Purpose

Subroutine LIQUID first determines the liquid storage capacity of the snowpack. It then determines how much liquid water is available to fill the storage capacity. Any liquid water above the capacity will leave the snowpack unless it freezes (see subroutine ICING).

Method

The liquid water holding capacity of the snowpack can be at the maximum as specified by the parameter MWATER, at zero, or somewhere in between depending on the density of the pack: the less dense the snowpack the greater the holding capacity. The following relationships define the capacity:

for $RDENPF > 0.91$,

$$PACKWC = 0.0 \qquad (19)$$

for $0.6 < RDENPF < 0.91$,

$$PACKWC = MWATER*(3.0 - 3.33*RDENPF) \qquad (20)$$

for RDENPF < 0.61,

$$PACKWC = MWATER \tag{21}$$

where:

PACKWC = liquid water holding capacity of the snowpack in in./in.
MWATER = parameter specifing the maximum liquid water content of the snowpack in in./in.
RDENPF = density of the snowpack relative to liquid water

MWATER is a function of the mass of ice layers, the size, the shape, and spacing of snow crystals and the degree of channelization and honeycombing of the snowpack.

Once PACKWC is calculated, it is compared to the available liquid water in the pack PWSUPY. PWSUPY is calculated by summing any storage remaining at the start of the interval, any melt, and any rain that fell on the pack which did not freeze. If PWSUPY is more than PACKWC, then water is yielded to the land surface from the snowpack.

4.2(1).2.10 Simulate Occurrence of Ice in the Pack (subroutine ICING)

Purpose

The purpose of subroutine ICING is to simulate the possible freezing of water which would otherwise leave the snowpack. This freezing in turn produces ice or frozen ground at the bottom of the snowpack. In this subroutine, the ice can be considered to be at the bottom of the pack or frozen in the ground below the snow portion of the pack thus extending the total pack into the soil. This subroutine may only be applicable in certain areas; therefore, it is user optional.

Method

The freezing of the water yield of the snowpack depends on the capacity of the environment to freeze it. Every day at approximately 6 a.m. the capacity is reassessed. A new value is estimated in terms of inches of melt by multiplying the Fahrenheit degrees of the air temperature below 32.0 by 0.01. This estimate is compared with the freezing capacity if any which remains from the previous 24-hr period. If it is greater, then the new estimated capacity replaces the old, else the old value remains as the potential. Any water yield that occurs freezes and is added to the ice portion of the snowpack until the capacity is met. Any subsequent water yield is released from the snowpack.

4.2(1).2.11 Melt the Pack Using Heat from the Ground (subroutine GMELT)

Purpose

The purpose of the GMELT subroutine is to simulate the melt caused by heat conducted from the surface underlying the snowpack. This ground heat melts the pack only from below. Therefore, melt from this process is considered independent of other previously calculated heat influences except for an indirect effect via the temperature of the snowpack. Unlike the other melt processes, ground heat melts the ice portion of the snowpack first since ice is considered to be located at the lower depths of the pack.

Method

The potential rate of ground melt is calculated hourly as a function of snowpack temperature (PAKTMP) and a lumped parameter (MGMELT). MGMELT is the maximum rate of melt in water equivalent caused by heat from the ground at a PAKTMP of 32 degrees F. MGMELT would depend upon the thermal conductivity of the soil and the normal depth of soil freezing. The potential ground melt is reduced below MGMELT by 3 percent for each degree that PAKTMP is below 32 degrees F to a minimum of 19 percent of MGMELT at 5 degrees F or lower. As long as a snowpack is present, ground melt occurs at this potential rate.

4.2(1).2.12 Reset State Variables When Snowpack Disappears (subroutine NOPACK)

Purpose

This code resets the state variables (for example, SNOCOV) when the snowpack completely disappears.

Method

The frozen contents of the snowpack required for complete areal cover of snow (COVINX) is set to a tenth of the maximum value (COVIND). All other variables are either set to zero or the "undefined" value of -1.0E38.

4.2(1).3 Simulate Water Budget for a Pervious Land Segment (Section PWATER of Module PERLND)

Purpose

PWATER is used to calculate the components of the water budget, primarily to predict the total runoff from a pervious area. PWATER is the key component of module PERLND; subsequent major sections of PERLND (eg. SEDMNT) depend on the outputs of this section.

Background

The hydrologic processes that are modeled by PWATER are illustrated in Figure 4.2(1).3-1. The algorithms used to simulate these land related processes are the product of over 15 yr of research and testing. They are based on the original research for the LANDS subprogram of the Stanford Watershed Model IV (Crawford and Linsley 1966). LANDS has been incorporated into many models and used to successfully simulate the hydrologic responses of widely varying watersheds. The equations used in module section PWATER are nearly identical to the ones in the current version of LANDS in the PTR Model (Crawford and Donigian 1973), HSP (Hydrocomp 1976), and the ARM and NPS Models (Donigian and Crawford 1976 a,b). However, some changes have been made to LANDS to make the algorithms internally more amenable to a range of calculation time steps. Also, many of the parameter names have been changed to make them more descriptive, and some can be input on a monthly basis to allow for seasonal variations.

Data Requirements and Manipulation

The number of time series required by module section PWATER depends on whether snow accumulation and melt are considered.

When such conditions are not considered, only potential evapotranspiration and precipitation are required.

However, when snow conditions are considered, air temperature, rainfall, snowcover, water yield, and ice content of the snowpack are also required. Also, the evaporation data are adjusted when snow is considered. The input evaporation values are reduced to account for the fraction of the land segment covered by the snowpack (determined from the generated time series for snow cover), with an allowance for the fraction of area covered by coniferous forest which, it is assumed, can transpire through any snow cover. Furthermore, PET is reduced to zero when air temperature is below the parameter PETMIN. If air temperature is below PETMAX but above PETMIN, PET will be reduced to 50% of the input value, unless the first adjustment already reduced it to less than this amount.

The estimated potential evapotranspiration (PET) is used to calculate actual ET in subroutine group EVAPT.

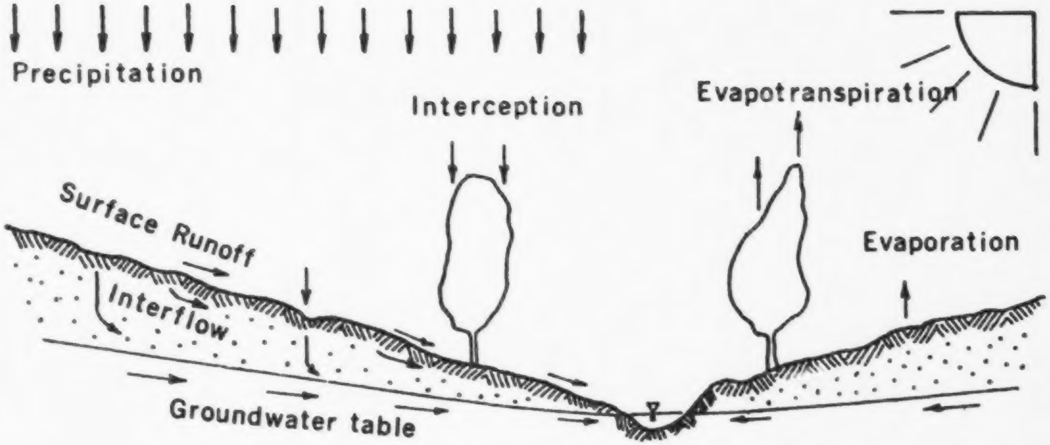


Figure 4.2(1).3-1 Hydrologic cycle

Approach

Figure 4.2(1).3-2 represents the fluxes and storages simulated in module section PWATER. The time series SUPY representing moisture supplied to the land segment includes rain, and when snow conditions are considered, rain plus water from the snowpack. SUPY is then available for interception. Interception storage is water retained by any storage above the overland flow plane. For pervious areas, interception storage is mostly on vegetation. Any overflow from interception storage is added to the optionally supplied time series of surface external lateral inflow to produce the total inflow into the surface detention storage.

Inflow to the surface detention storage is added to existing storage to make up the water available for infiltration and runoff. Moisture which directly infiltrates moves to the lower zone and groundwater storages. Other water may go to the upper zone storage, may be routed as runoff from surface detention or interflow storage, or may stay on the overland flow plane, from which it runs off or infiltrates at a later time.

The processes of infiltration and overland flow interact and occur simultaneously in nature. Surface conditions such as heavy turf on mild slopes restrict the velocity of overland flow and reduce the total quantity of runoff by allowing more time for infiltration. Increased soil moisture due to prolonged infiltration will in time reduce the infiltration rate producing more overland flow. Surface detention will modify flow. For example, high intensity rainfall is attenuated by storage and the maximum outflow rate is reduced. The water in the surface detention may also later infiltrate reoccurring as interflow, or it can be contained in upper zone storage.

Water infiltrating through the surface and percolating from the upper zone storage to the lower zone storage may flow to active groundwater storage or may be lost by deep percolation. Active groundwater eventually reappears as baseflow, but deep percolation is considered lost from the simulated system.

Lateral external inflows to interflow and active groundwater storages are also possible in section PWATER. One may wish to use this option if an upslope land segment is significantly different to merit separating it from a downslope land segment and no channel exists between them. This capability was not included in the previous models.

Not only are flows important in the simulation of the water budget, but so are storages. As stated, soil storage affects infiltration. The water holding capacity of the two soil storages, upper zone and lower zone, in module section PERLND is defined in terms of nominal capacities. Nominal, rather than absolute capacities, serve the purpose of smoothing any abrupt change that would occur if an absolute capacity is reached. Such capacities permit a smooth transition in hydrologic performance as the water content fluctuates.

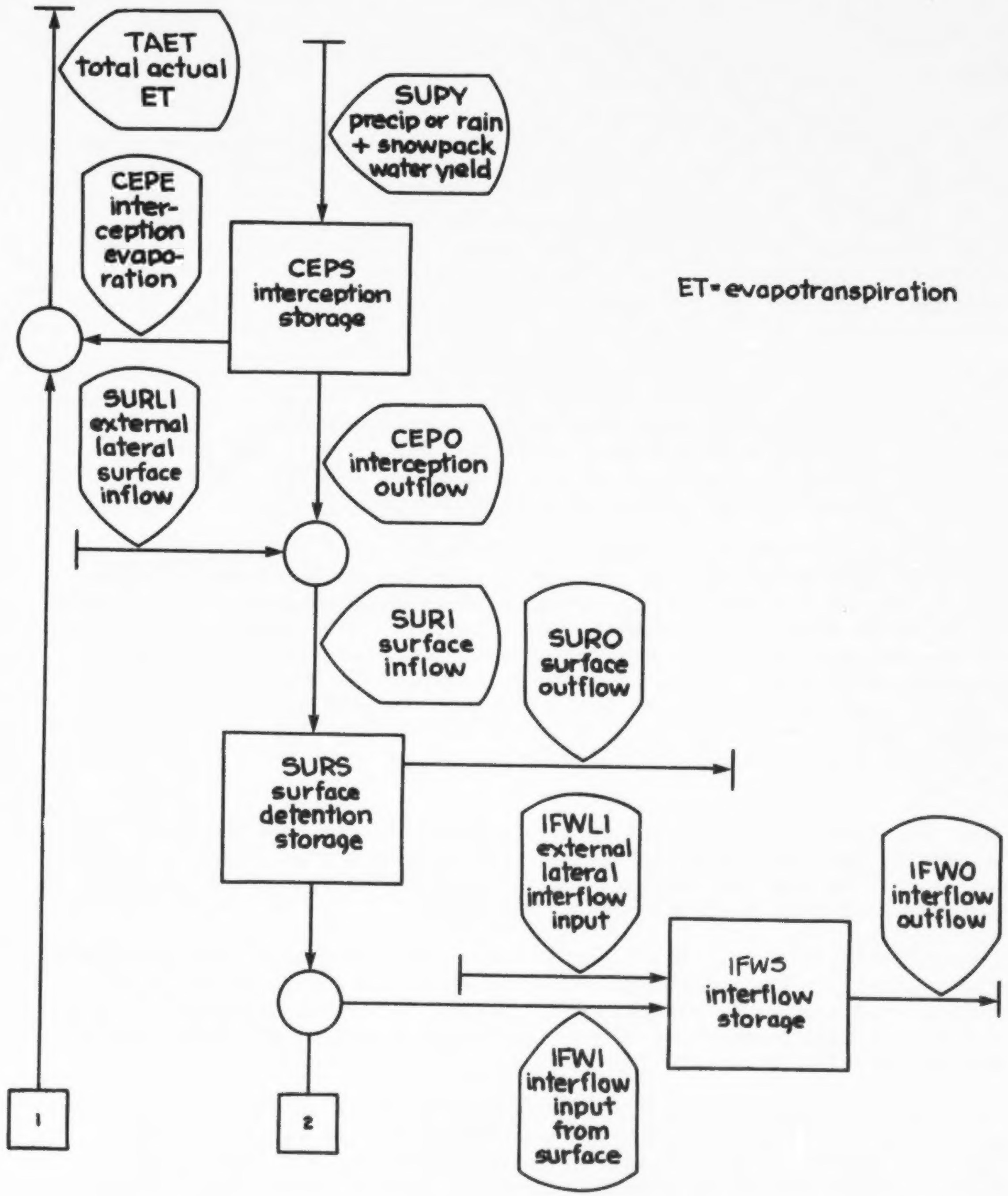


Figure 4.2(1).3-2 Flow diagram of water movement and storages modeled in the PWATER section of the PERLND Application Module.

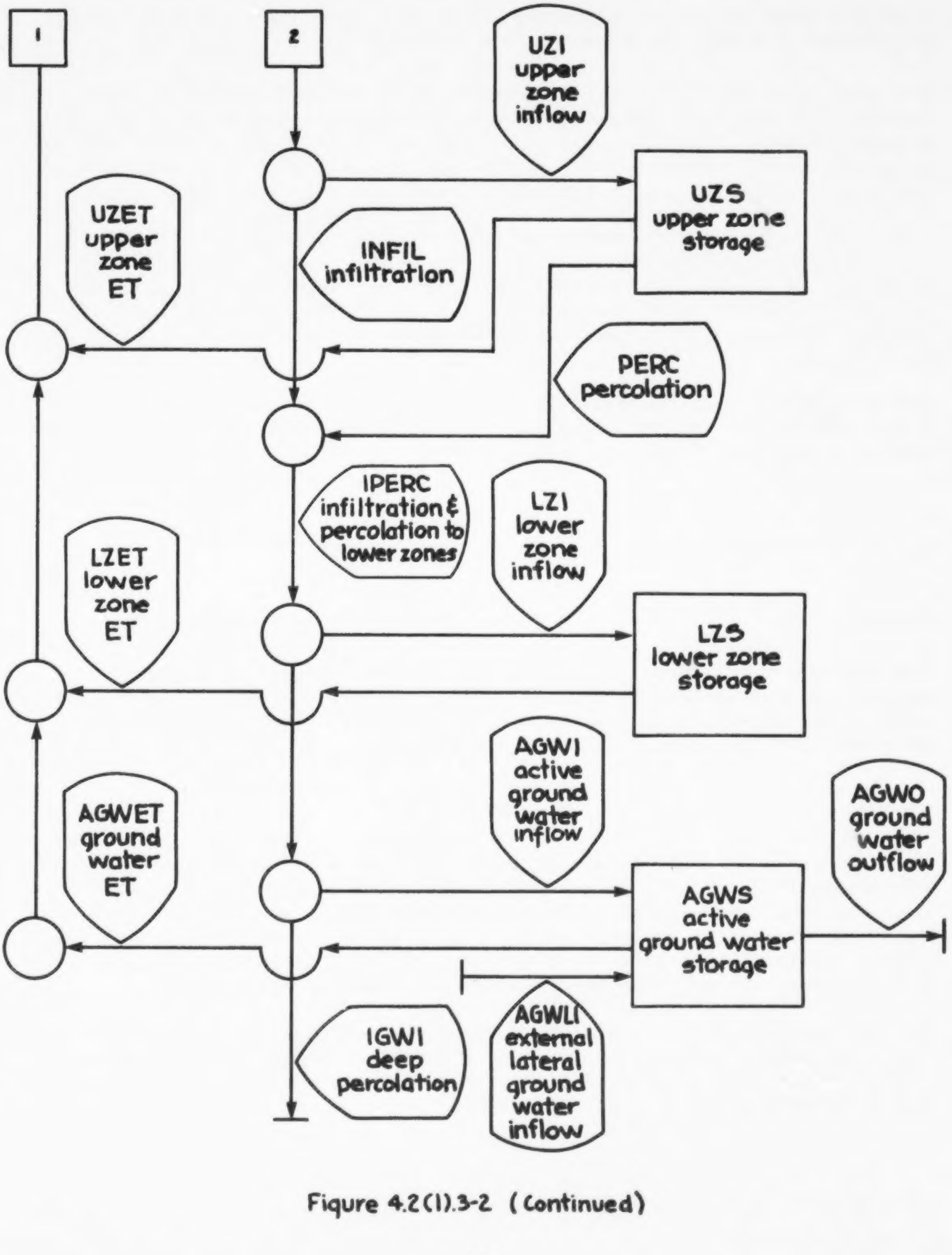


Figure 4.2(1).3-2 (Continued)

Storages also affect evapotranspiration loss. Evapotranspiration can be simulated from interception storage, upper and lower zone storages, active groundwater storage, and directly from baseflow.

Storages and flows can also be instrumental in the transformation and movement of chemicals simulated in the agri-chemical module sections. Soil moisture levels affect the adsorption and transformations of pesticides and nutrients. Soil moisture contents may vary greatly over a land segment. Therefore, a more detailed representation of the moisture contents and fluxes may be needed to simulate the transport and reaction of agricultural chemicals. Following the ARM Model, HSPF permits a segment to be further divided into conceptual "blocks" which represent the areal variations of the watershed in more detail. Further explanation of these conceptual areal blocks or zones is given in the ARM Model report (Donigian and Crawford 1976a). ARM uses five blocks but HSPF allows the user to specify from one to five.

The following subroutine descriptions will explain in more detail the algorithms of the PWATER module section. Further detail can be found in the pseudo code and the reports cited above.

4.2(1).3.1 Simulate Interception (subroutine ICEPT)

Purpose

The purpose of this code is to simulate the interception of moisture by vegetal or other ground cover. Moisture is supplied by precipitation, or under snow conditions, it is supplied by the rain not falling on the snowpack plus the water yielded by the snowpack.

Method

The user may supply the interception capacity on a monthly basis to account for seasonal variations, or may supply one value designating a fixed capacity. The interception capacity parameter can be used to designate any retention of moisture which does not infiltrate or reach the overland flow plane. Typically for pervious areas this capacity represents storage on grass blades, leaves, branches, trunks, and stems of vegetation.

Moisture exceeding the interception capacity overflows the storage and is ready for either infiltration or runoff as determined by subroutine group SURFAC. Water held in interception storage is removed by evaporation; the amount is determined in subroutine EVICEP.

4.2(1).3.2 Distribute the Water Available for Infiltration and Runoff (subroutine SURFAC)

Purpose

Subroutine SURFAC determines what happens to the moisture on the surface of the land. It may infiltrate, go to the upper zone storage or interflow storage, remain in surface detention storage, or run off.

Method

The algorithms which simulate infiltration represent both the continuous variation of infiltration rate with time as a function of soil moisture and the areal variation of infiltration over the land segment. The equations representing the dependence of infiltration on soil moisture are based on the work of Philip (1957) and are derived in detail in the previously cited reports.

The infiltration capacity, the maximum rate at which soil will accept infiltration, is a function of both the fixed and variable characteristics of the watershed. Fixed characteristics include primarily soil permeability and land slopes, while variables are soil surface conditions and soil moisture content. Fixed and variable characteristics vary spatially over the land segment. A linear probability density function is used to account for areal variation. Figure 4.2(1).3-3 represents the infiltration/interflow/surface runoff distribution function of section PWATER. Careful attention to this figure and Figure 4.2(1).3-2 will facilitate understanding of subroutine SURFAC and the subordinate subroutines DISPOS, DIVISN, UZINF, and PROUTE.

The infiltration distribution represented by Figure 4.2(1).3-3 is focused around the two lines which separate the moisture available to the land surface (MSUPY) into what infiltrates and what goes to interflow. A number of the variables that are used to determine the location of lines I and II are calculated in subroutine SURFAC. They are calculated by the following relationships:

```
IBAR = (INFILT/(LZS/LZSN)**INFEXP)*INFFAC                      (1)

IMAX = INFILD*IBAR                                             (2)

IMIN = IBAR - (IMAX - IBAR)                                    (3)

RATIO = INTFW*(2.0**(LZS/LZSN))                                (4)
```

where:

IBAR = mean infiltration capacity over the land segment in in./interval

INFILT = infiltration parameter in in./interval

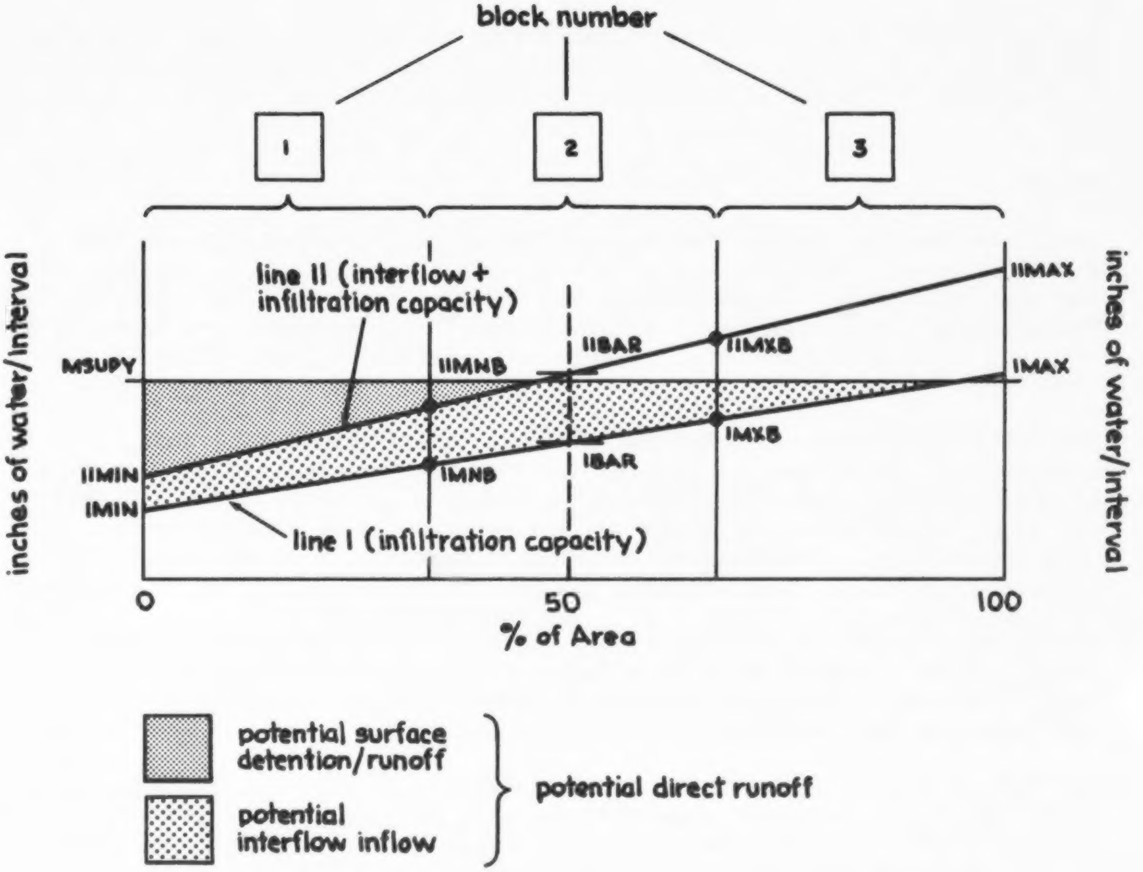


Figure 4.2(1).3-3 Determination of infiltration and interflow inflow

LZS = lower zone storage in inches
LZSN = parameter for lower zone nominal storage in inches
INFEXP = exponent parameter greater than one
INFFAC = factor to account for frozen ground effects, if applicable
IMAX = maximum infiltration capacity in in./interval
INFILD = parameter giving the ratio of maximum to mean infiltration capacity over the land segment
IMIN = minimum infiltration capacity in in./interval
RATIO = ratio of the ordinates of line II to line I
INTFW = interflow inflow parameter

The factor that reduces infiltration (and also upper zone percolation) to account for the freezing of the ground surface (INFFAC) is 1.0 if icing is not simulated. When icing occurs, the factor is 1.0 minus the water equivalent of ice in of the snowpack to a minimum of 0.1.

The parameter INTFW can be input on a monthly basis to allow for variations throughout the year.

When the land segment is separated into conceptual areal blocks as designated by the vertical subdivisions diagrammed in Figure 4.2(1).3-3, corresponding IMAX and IMIN values must be determined for each block:

$$IMNB = IMIN + (BLK - 1)*(IMAX - IMIN)/NBLKS \tag{5}$$

$$IMXB = IMNB + (IMAX - IMIN)/NBLKS \tag{6}$$

where:
IMNB = minimum infiltration capacity for block BLK in in./interval
BLK = block number
NBLKS = total number of blocks being simulated
IMXB = maximum infiltration capacity for block BLK in in./interval

4.2(1)3.2.1 Dispose of Moisture Supply (subroutine DISPOS)

Purpose

Subroutine DISPOS determines what happens to the moisture supply (MSUPY) on either an individual block of the land segment (if NBLKS is greater than 1) or on the entire land segment (if NBLKS is equal to 1).

Method

This subroutine calls subordinate routines DIVISN, UZINF, and PROUTE. DIVISN is called to determine how much of MSUPY falls above and below line I in Figure 4.2(1).3-3. The quantity under this line is considered to be infiltrated. The amount over the line but under the MSUPY line (the entire shaded portion) is the potential direct runoff (PDRO), which is the combined increment to interflow, and upper zone storage plus the quantities which will stay on the surface and run off. PDRO is subdivided by line II. The ordinates of line II are found by multiplying the ordinates of line I by RATIO (see subroutine SURFAC for definition). The quantity underneath both line II and the MSUPY line but above line I is called potential interflow inflow. This consists of actual interflow plus an increment to upper zone storage. Any amount above line II but below the MSUPY (potential surface detention/runoff) is that portion of the moisture supply which stays on the surface and is available for overland flow routing, plus a further increment to upper zone storage. The fractions of the potential interflow inflow and potential surface detention/runoff which are combined to compose the upper zone inflow are determined in subroutine UZINF.

4.2(1).3.2.1.2 Compute Inflow to Upper Zone (subroutines UZINF1 and UZINF2)

Purpose

The purpose of this code is to compute the inflow to the upper zone when there is some potential direct runoff (PDRO). PDRO, which is determined in subroutine DISPOS, will either enter the upper zone storage or be available for either interflow or overland flow. This subroutine determines what amount, if any, will go to the upper zone storage.

Method

The fraction of the potential direct runoff which is inflow to the upper zone storage is a function of the ratio (UZRAT) of the storage to the nominal capacity. Figure 4.2(1).3-4 diagrams this relationship. The equations used to define this curve follow:

$$\text{FRAC} = 1 - (\text{UZRAT}/2)*(1/(4 - \text{UZRAT}))**(3 - \text{UZRAT}) \tag{7}$$

for UZRAT less than or equal to two. For UZRAT greater than two,

$$\text{FRAC} = (0.5/(\text{UZRAT}-1))**(2*\text{UZRAT}-3) \tag{8}$$

where:

FRAC = fraction of PDRO retained by the upper zone storage
UZRAT = UZS/UZSN

Since UZS and FRAC are dynamically affected by the inflow process it becomes desirable when using particularly large time steps to integrate over the interval to find the inflow to the upper zone. This is done in subroutine UZINF1. The solution is simplified by assuming that inflow to and outflow from the upper zone is handled separately. Considering inflow, the following differential equation results:

$$d(UZS)/dt = (d(UZRAT)/dt)*UZSN = PDRO*FRAC \qquad (9)$$

Thus

$$d(UZRAT)/FRAC = (PDRO/UZSN)*dt \qquad (10)$$

Now taking the definite integral of both sides of the equation:

$$INTGRL = \int_{UZRAT_{t1}}^{UZRAT_{t2}} \frac{d(UZRAT)}{FRAC} = (PDRO/UZSN)(t2-t1) \qquad (11)$$

where:

t1 = time at start of interval
t2 = time at end of interval

The integral on the left side must be evaluated numerically. Subroutine UZINF1 uses tabulated corresponding values of INTGRL and UZRAT to evaluate it. This relationship, plus Equations 9 and 11, enable one to find the change in UZRAT over the interval and, hence, the quantity of inflow.

Subroutine UZINF2, which is an alternative to UZINF1, uses the same algorithm as HSP, ARM and NPS. That is, Equations 7 and 8 are used directly to estimate the fraction of PDRO retained by the upper zone. Only the value of UZRAT at the start of the simulation interval is used; thus, no account is taken of the possible steady reduction in inflow to the upper zone within a single time step, due to its being filled (Figure 4.2(1).3-4).

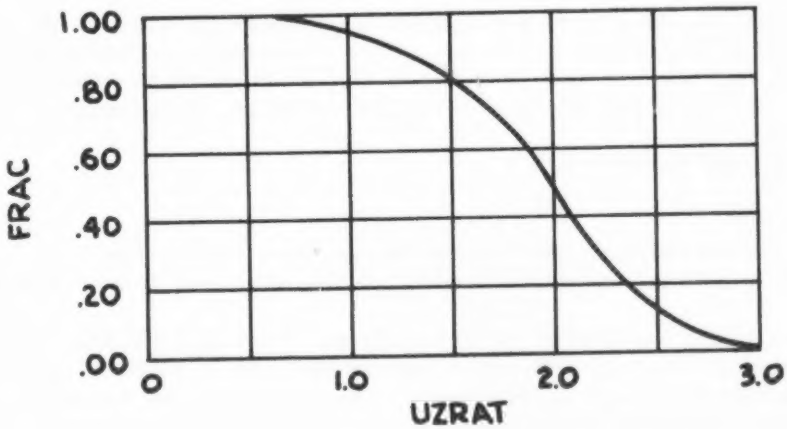


Figure 4.2(1).3-4 Fraction of the potential direct runoff retained by the upper zone (FRAC) as a function of the upper zone soil moisture ratio (UZRAT)

4.2(1).3.2.1.3 Determine Surface Runoff (subroutine PROUTE)

Purpose

The purpose of subroutine PROUTE is to determine how much potential surface detention runs off in one simulation interval.

Method of Routing

Overland flow is treated as a turbulent flow process. It is simulated using the Chezy-Manning equation and an empirical expression which relates outflow depth to detention storage. A more detailed explanation and derivation can be found in the reports cited in the initial background discussion. The rate of overland flow discharge is determined by the equations:

for SURSM < SURSE
    SURO = DELT60*SRC*(SURSM*(1.0 + 0.6(SURSM/SURSE)**3)**1.67 (12)

for SURSM >= SURSE
    SURO = DELT60*SRC*(SURSM*1.6)**1.67

where:
SURO = surface outflow in in./interval
DELT60 = DELT/60.0 (hr/interval)
SRC = routing variable, described below
SURSM = mean surface detention storage over the time interval in inches
SURSE = equilibrium surface detention storage (inches) for current supply rate

DELT60 makes the equations applicable to a range of time steps (DELT). The first equation represents the case where the overland flow rate is increasing, and the second case where the surface is at equilibrium or receding. Equilibrium surface detention storage is calculated by:

SURSE = DEC*SSUPR**0.6 (13)

where:
DEC = calculated routing variable, described below
SSUPR = rate of moisture supply to the overland flow surface

There are two optional ways of determining SSUPR and SURSM. One option - the same method used in the prior models HSP,ARM and NPS - estimates SSUPR by subtracting the surface storage at the start of the interval (SURS) from the potential surface detention (PSUR) which was determined in subroutine DISPOS. The units of SSUPR are inches per interval. SURSM is estimated as the mean of SURS and PSUR. The other option estimates SSUPR by the same method except that the result is divided by DELT60 to obtain a value with units of inches per hour. SURSM is set equal to SURS. This option has not been used in prior models, but is dimensionally consistent for any time step.

The variables DEC and SRC are calculated daily in subroutine SURFAC, but their equations will be given here since they pertain to routing. They are:

DEC = 0.00982*(NSUR*LSUR/SQRT(SLSUR))**0.6 (14)

SRC = 1020.0*(SQRT(SLSUR)/(NSUR*LSUR)) (15)

where:
NSUR = Manning's n for the overland flow plane
LSUR = length of the overland flow plane in ft
SLSUR = slope of the overland flow plane in ft/ft

NSUR can be input on a monthly basis to allow for variations in roughness of the overland flow plane throughout the year.

4.2(1).3.3 Simulate Interflow (subroutine INTFLW)

Purpose

Interflow can have an important influence on storm hydrographs particularly when vertical percolation is retarded by a shallow, less permeable soil layer. Additions to the interflow component are retained in storage or routed as outflow from the land segment. Inflows to the interflow component may occur from the surface or from upslope external lateral flows. The purpose of this subroutine is to determine the amount of interflow and to update the storage.

Method of Determining Interflow

The calculation of interflow outflow assumes a linear relationship to storage. Thus outflow is a function of a recession parameter, inflow, and storage. Moisture that remains will occupy interflow storage. Interflow discharge is calculated by:

IFWO = (IFWK1*INFLO) + (IFWK2*IFWS) (16)

where:
IFWO = interflow outflow in in./interval
INFLO = inflow into interflow storage in in./interval
IFWS = interflow storage at the start of the interval in inches

IFWK1 and IFWK2 are variables determined by:

IFWK1 = 1.0 - (IFWK2/KIFW) (17)

IFWK2 = 1.0 - EXP(-KIFW) (18)

and

KIFW = -ALOG(IRC)*DELT60/24.0 (19)

where:

IRC = interflow recession parameter, per day
DELT60 = number of hr/interval
24.0 = number of hours per day
EXP = Fortran exponential function
ALOG = fortran natural logarithm function

When a pervious land segment is divided into more than one block, the algorithms are applied separately to each block. IRC is the ratio of the present rate of interflow outflow to the value 24 hours earlier, if there was no inflow. IRC can be input on a monthly basis to allow for variations in soil properties throughout the year.

4.2(1).3.4 Simulate Upper Zone Behavior (subroutine UZONE)

Purpose

This subroutine and the subsidiary subroutine UZONES are used to calculate the water percolating from the upper zone. Water not percolated remains in upper zone storage available for evapotranspiration in subroutine ETUZON.

Method of Determining Percolation

The upper zone inflow calculated in DISPOS is first added to the upper zone storage at the start of the interval to obtain the total water available for percolation from the upper zone.

Percolation only occurs when UZRAT minus LZRAT is greater than 0.01. When this happens, percolation from the upper zone storage is calculated by the empirical expression:

PERC = 0.1*INFILT*INFFAC*UZSN*(UZRAT - LZRAT)**3 (20)

where:

PERC = percolation from the upper zone in in./interval
INFILT = infiltration parameter in in./interval
INFFAC = factor to account for frozen ground, if any,
UZSN = parameter for upper zone nominal storage in inches
UZRAT = ratio of upper zone storage to UZSN
LZRAT = ratio of lower zone storage to lower zone nominal storage (LZSN)

The upper zone nominal capacity can be input on a monthly basis to allow for variations throughout the year. The monthly values are interpolated to obtain daily values. When a pervious land segment is divided into more than one block, the algorithm is applied separately to each block.

4.2(1).3.5 Simulate Lower Zone Behavior (subroutine LZONE)

Purpose

This subroutine determines the quantity of infiltrated and percolated water which enters the lower zone. The infiltrated moisture supply is determined in subroutine DISPOS. The percolated moisture from the upper zone is found in subroutine UZONE.

Method

The fraction of the direct infiltration plus percolation that enters the lower zone storage (LZS) is based on the lower zone storage ratio of LZS/LZSN where LZSN is the lower zone nominal capacity. The inflowing fraction is determined empirically by:

$$\mathrm{LZFRAC} = 1.0 - \mathrm{LZRAT}*(1.0/(1.0 + \mathrm{INDX}))**\mathrm{INDX} \tag{21}$$

when LZRAT is less than 1.0, and by

$$\mathrm{LZFRAC} = (1.0/(1.0 + \mathrm{INDX}))**\mathrm{INDX} \tag{22}$$

when LZRAT is greater than 1.0. INDX is defined by:

$$\mathrm{INDX} = 1.5*\mathrm{ABS}(\mathrm{LZRAT} - 1.0) + 1.0 \tag{23}$$

where:
LZFRAC = fraction of infiltration plus percolation entering LZS
LZRAT = LZS/LZSN
ABS = function for determining absolute value

These relationships are plotted in Figure 4.2(1).3-5. The fraction of the moisture supply remaining after the surface, upper zone, and lower zone components are subtracted is added to the groundwater storages.

4.2(1).3.6 Simulate Groundwater Behavior (subroutine GWATER)

Purpose

The purpose of this subroutine is to determine the amount of the inflow to groundwater that is lost to deep or inactive groundwater and to determine the amount of active groundwater outflow. These two fluxes will in turn affect the active groundwater storage.

Method of Determining Groundwater Fluxes

The quantity of direct infiltration plus percolation from the upper zone which does not go to the lower zone (determined in subroutine LZONE) will be inflow to either inactive or active groundwater. The distribution to active

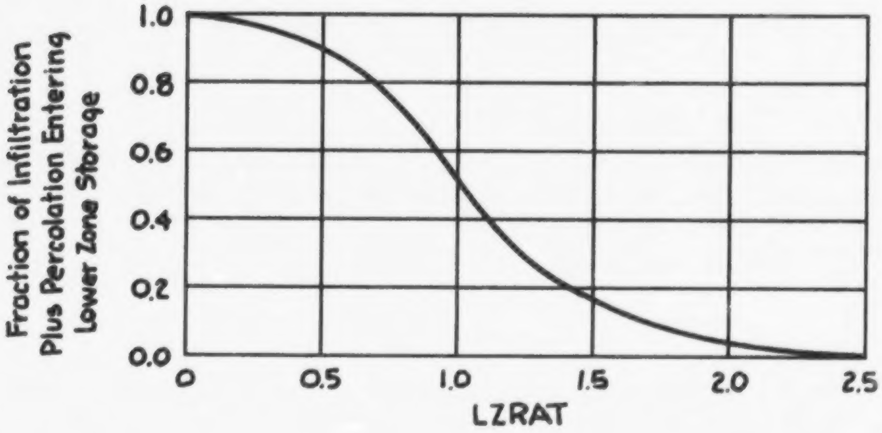


Figure 4.2(1).3-5 Fraction of infiltration plus percolation entering lower zone storage

and inactive groundwater is user designated by parameter DEEPFR. DEEPFR is that fraction of the groundwater inflow which goes to inactive groundwater. The remaining portion of the percolating water and all external lateral inflow if any make up the total inflow to the active groundwater storage.

The outflow from active groundwater storage is based on a simplified model. It assumes that the discharge of an aquifer is proportional to the product of the cross-sectional area and the energy gradient of the flow. Further, a representative cross-sectional area of flow is assumed to be related to the groundwater storage level at the start of the interval. The energy gradient is estimated as a basic gradient plus a variable gradient that depends on past active groundwater accretion.

Thus, the groundwater outflow is estimated by:

$$AGWO = KGW*(1.0 + KVARY*GWVS)*AGWS \tag{24}$$

where:

AGWO = active groundwater outflow in in./interval
KGW = groundwater outflow recession parameter, per interval
KVARY = parameter which can make active groundwater storage to outflow relation nonlinear in per inches
GWVS = index to groundwater slope in inches
AGWS = active groundwater storage at the start of the interval in inches

GWVS is increased each interval by the inflow to active groundwater but is also decreased by 3 percent once a day. It is a measure of antecedent active groundwater inflow. KVARY is introduced to allow variable groundwater recession rates. When KVARY is nonzero, a semilog plot of discharge vs. time is nonlinear. This parameter adds flexibility in groundwater outflow simulation which is useful in simulating many watersheds.

The parameter KGW is calculated by the Run Interpreter using the relationship:

$$KGW = 1.0 - (AGWRC)^{**}(DELT60/24.0) \qquad (25)$$

where:

AGWRC = daily recession constant of groundwater flow, if KVARY or GWVS = 0.0
That is, the ratio of current groundwater discharge to groundwater discharge 24-hr earlier

DELT60 = hr/interval

4.2(1).3.7 Simulate Evapotranspiration (subroutine EVAPT)

Purpose

The purpose of EVAPT and its subordinate subroutines is to simulate evaporation and evapotranspiration fluxes from all zones of the pervious land segment. Since in most hydrologic regimes the volume of water that leaves a watershed as evapotranspiration exceeds the total volume of streamflow, this is an important aspect of the water budget.

Method of Determining Actual Evapotranspiration

There are two separate issues involved in estimating evapotranspiration (ET). First, potential ET must be estimated. ET potential or demand is supplied as an input times series, typically using U.S. Weather Bureau Class A pan records plus an adjustment factor. The data are further adjusted for cover and temperature in the parent subroutine PWATER. Second, actual ET must be calculated, usually as a function of moisture storages and the potential. The actual ET is estimated by trying to meet the demand from five sources in the order described below. The sum of the ET from these five sources is the total actual evapotranspiration from the land segment.

Subroutine ETBASE

The first source from which ET can be taken is the active groundwater outflow or baseflow. This simulates effects such as ET from riparian vegetation in which groundwater is withdrawn as it enters the stream. The user may specify by the parameter BASETP the fraction, if any, of the potential ET that can be sought from the baseflow. That portion can only be fulfilled if outflow exists. Any remaining potential not met by actual baseflow evaporation will try next to be satisfied in subroutine EVICEP.

Subroutine EVICEP

Remaining potential ET then exerts its demand on the water in interception storage. Unlike baseflow, there is no parameter regulating the rate of ET from interception storage. The demand will draw upon all of the interception storage unless the demand is less than the storage. When the demand is greater than the storage, the remaining demand will try to be satisfied in subroutine ETUZON.

Subroutine ETUZON

There are no special ET parameters for the upper zone, but rather ET is based on the moisture in storage in relation to its nominal capacity. Actual evapotranspiration will occur from the upper zone storage at the remaining potential demand if the ratio of UZS/UZSN, upper zone storage to nominal capacity, is greater than 2.0. Otherwise the remaining potential ET demand on the upper zone storage is reduced; the adjusted value depends on UZS/UZSN. Subroutine ETAGW will attempt to satisfy any remaining demand.

Subroutine ETAGW

Like ET from baseflow, actual evapotranspiration from active groundwater is regulated by a parameter. The parameter AGWETP is the fraction of the remaining potential ET that can be sought from the active groundwater storage. That portion of the ET demand can be met only if there is enough active groundwater storage to satisfy it. Any remaining potential will try to be met in subroutine ETLZON.

Subroutine ETLZON

The lower zone is the last storage from which ET is drawn. Evapotranspiration from the lower zone is more involved than that from the other storages. ET from the lower zone depends upon vegetation transpiration. Evapotranspiration opportunity will vary with the vegetation type, the depth of rooting, density of the vegetation cover, and the stage of plant growth along with the moisture characteristics of the soil zone. These influences on the ET opportunity are lumped into the LZETP parameter. Unlike the other ET parameters LZETP can be input on a monthly basis to account for temporal changes in the above characteristics.

If the LZETP parameter is at its maximum value of one, representing near complete areal coverage of deep rooted vegetation, then the potential ET for the lower zone is equal to the demand that remains. However, this is normally not the case. Usually vegetation type and/or rooting depths will vary over the land segment. To simulate this, a linear probability density function for ET opportunity is assumed (Figure 4.2(1).3-6). This approach is similar to that used to handle areal variations in infiltration/percolation capacity.

The variable RPARM, the index to maximum ET opportunity, is estimated by:

RPARM = 0.25/(1.0 - LZETP)*(LZS/LZSN)*DELT60/24.0 (26)

where:

RPARM = maximum ET opportunity in in./interval
LZETP = lower zone ET parameter
LZS = current lower zone storage in inches
LZSN = lower zone nominal storage parameter in inches
DELT60 = hr/interval

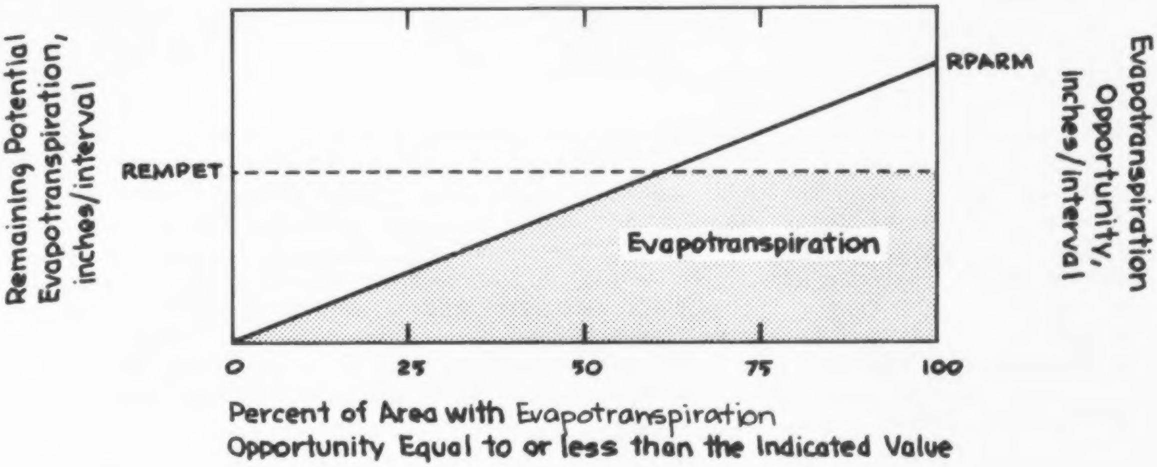


Figure 4.2(1).3-6 Potential and actual evapotranspiration rom the lower zone

The quantity of water lost by ET from the lower zone storage, when remaining potential ET (REMPET) is less than RPARM, is given by the cross-hatched area of Figure 4.2(1).3-6. When REMPET is more than RPARM the lower zone ET is equal to the entire area under the triangle, RPARM/2.

ET from the lower zone storage is further reduced when LZETP is less than 0.5 by multiplying by LZETP*2.0. This is designed to account for the fraction of the land segment devoid of any vegetation that can draw from the lower zone.

4.2(1).4 Simulate Production and Removal of Sediment (Section SEDMNT of Module PERLND)

Purpose

Module section SEDMNT simulates the production and removal of sediment from a pervious land segment. Sediment can be considered to be inorganic, organic, or both; the definition is up to the user.

Sediment from the land surface is one of the most common pollutants of waters from urban, agricultural, and forested lands. It can muddy waters, cover fish eggs, and limit the capacity of reservoirs. Nutritious and toxic chemicals can be carried by it.

Approach

The equations used to produce and remove sediment are based on the ARM and NPS Models (Donigian and Crawford 1976 a,b). The algorithms representing land surface erosion in these models were derived from a sediment model developed by Moshe Negev (Negev 1967) and influenced by Meyer and Wischmeier (1969) and Onstad and Foster (1975). The supporting management practice factor which has been added to the soil detachment by rainfall equation was based on the "P" factor in the Universal Soil Loss Equation (Wischmeier and Smith 1965). It was introduced in order to better evaluate agricultural conservation practices. The equation which represents the scouring of the matrix soil, which is not included in ARM or NPS, was derived from Negev's method for simulating gully erosion.

Figure 4.2(1).4-1 shows the detachment, attachment, and removal involved in the erosion processes on the pervious land surface, while Figure 4.2(1).4-2 schematically represents the fluxes and storages used to simulate these processes. Two of the sediment fluxes, SLSED and NSVI, are added directly to the detached sediment storage variable DETS in the parent subroutine SEDMNT while the other fluxes are computed in subordinate subroutines. SLSED represents external lateral input from an upslope land segment. It is a time series which the user may optionally specify. NVSI is a parameter that represents any net external additions or removals of sediment caused by human activities or wind.

Removal of sediment by water is simulated as washoff of detached sediment in storage (WSSD) and scour of matrix soil (SCRSD). The washoff process involves two parts: the detachment/attachment of sediment from/to the soil matrix and the transport of this sediment. Detachment (DET) occurs by rainfall. Attachment occurs only on days without rainfall; the rate of attachment is specified by parameter AFFIX. Transport of detached sediment is by overland flow. The scouring of the matrix soil includes both pick up and transport by overland flow combined into one process.

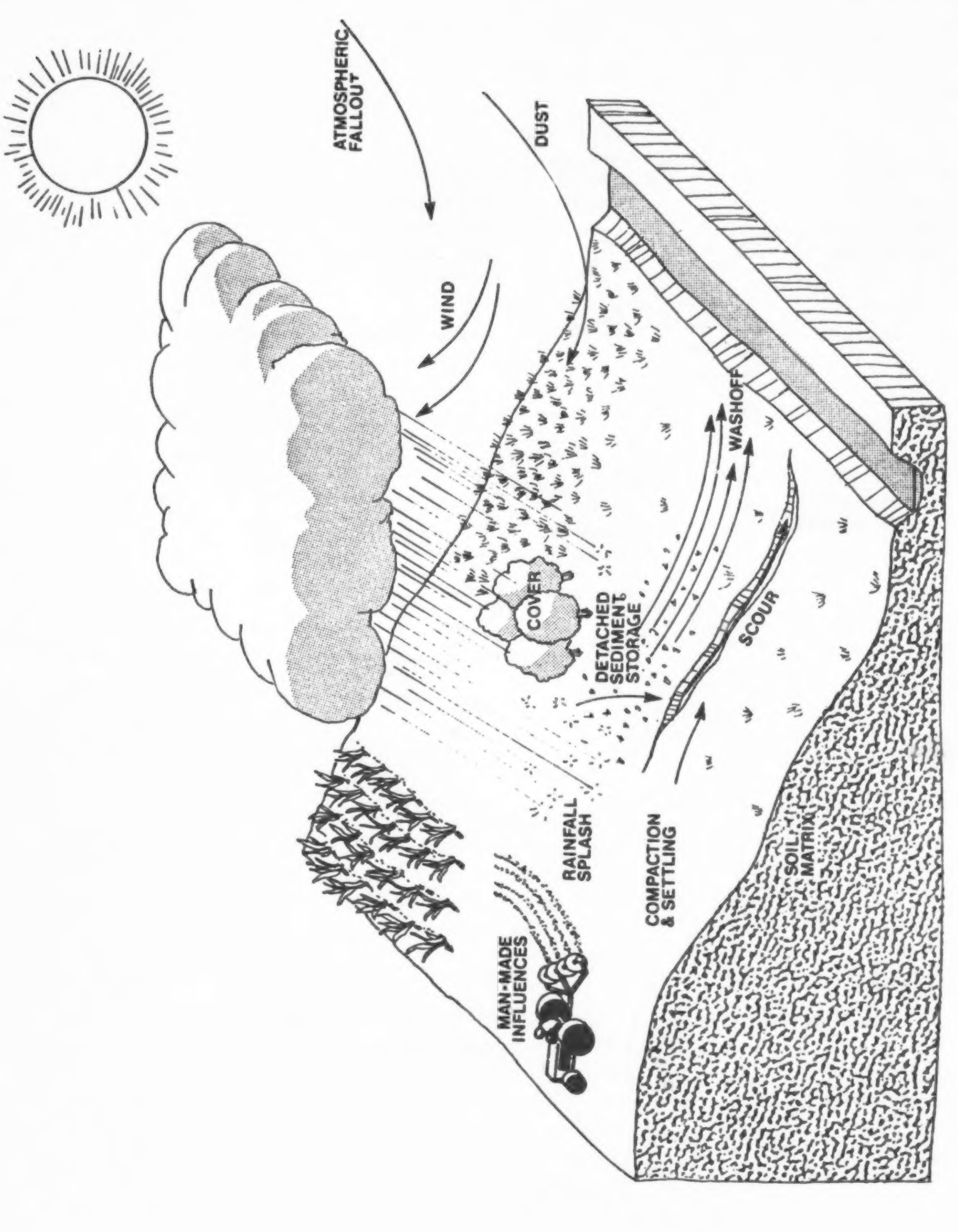


Figure 4.2(1).4-1 Erosion processes

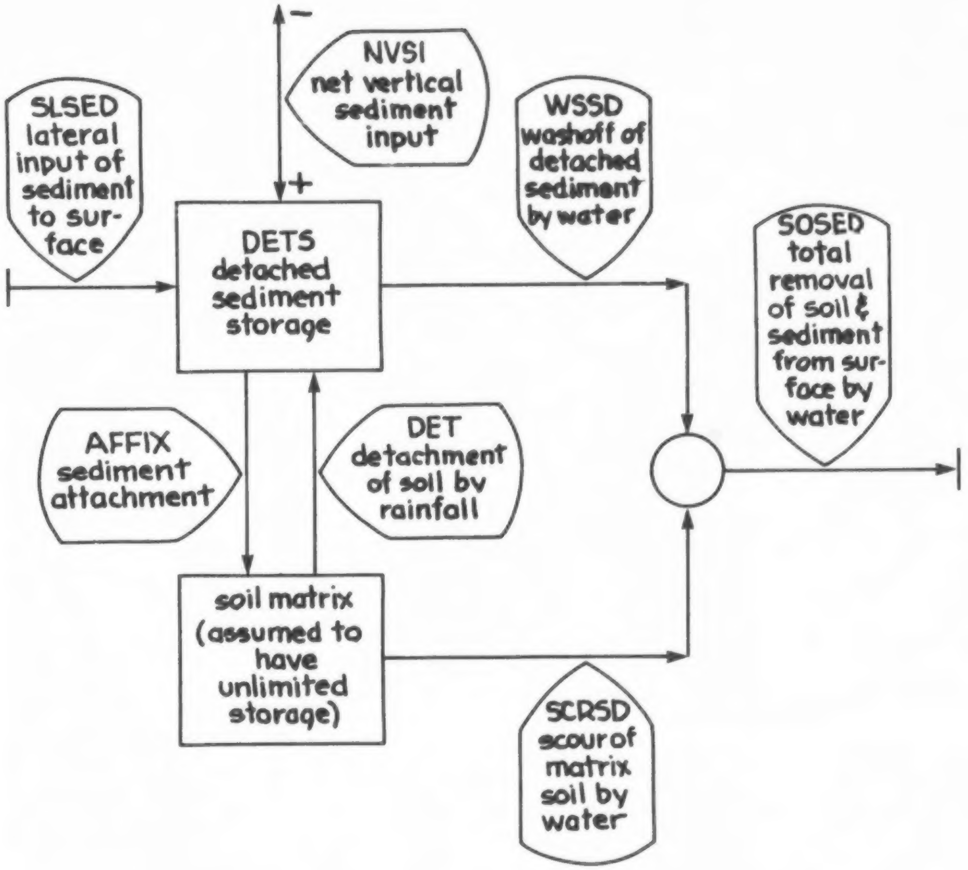


Figure 4.2(1).4-2. Flow diagram for SEDMNT section of PERLND Application Module.

Module section SEDMNT has two options for simulating washoff of detached sediment and scour of soil. One uses subroutine SOSED1 which is identical to the method used in the ARM and the NPS Models. However, some equations used in this method are dimensionally nonhomogeneous, and it has only been used with 15- and 5-min intervals. The results obtained are probably highly dependent on the simulation time step. The other option uses subroutine SOSED2 which is dimensionally homogeneous and is, theoretically, less dependent on the time step. However, it has not been tested.

4.2(1).4.1 Detach Soil By Rainfall (subroutine DETACH)

Purpose

The purpose of DETACH is to simulate the splash detachment of the soil matrix by falling rain.

Method of Detaching Soil by Rainfall

Kinetic energy from rain falling on the soil detaches particles which are then available to be transported by overland flow. The equation that simulates detachment is:

```
    DET    = DELT60*(1.0 - CR)*SMPF*KRER*(RAIN/DELT60)**JRER          (1)
```

where:

DET = sediment detached from the soil matrix by rainfall in tons/acre per interval
DELT60 = number of hr/interval
CR = fraction of the land covered by snow and other cover
SMPF = supporting management practice factor
KRER = detachment coefficient dependent on soil properties
RAIN = rainfall in in./interval
JRER = detachment exponent dependent on soil properties

The variable CR is the sum of the fraction of the area covered by the snowpack (SNOCOV), if any, and the fraction that is covered by anything else but snow (COVER). SNOCOV is computed by section SNOW. COVER is a parameter which for pervious areas will typically be the fraction of the area covered by vegetation and mulch. It can be input on a monthly basis.

4.2(1).4.2 Remove by Surface Flow Using Method 1 (subroutine SOSED1)

Purpose

Subroutines SOSED1 and SOSED2 perform the same task but by different methods. They simulate the washoff of the detached sediment and the scouring of the soil matrix.

Method

When simulating the washoff of detached sediment, the transport capacity of the overland flow is estimated and compared to the amount of detached sediment available. The transport capacity is calculated by the equation:

```
     STCAP = DELT60*KSER*((SURS + SURO)/DELT60)**JSER                (2)
where:
     STCAP  = capacity for removing detached sediment in
              tons/acre per interval
     DELT60 = hr/interval
     KSER   = coefficient for transport of detached sediment
     SURS   = surface water storage in inches
     SURO   = surface outflow of water in in./interval
     JSER   = exponent for transport of detached sediment
```

When STCAP is greater than the amount of detached sediment in storage, washoff is calculated by:

```
     WSSD = DETS*SURO/(SURS + SURO)                                  (3)
```

If the storage is sufficient to fulfill the transport capacity, then the following relationship is used:

```
     WSSD = STCAP*SURO/(SURS + SURO)                                 (4)
where:
     WSSD = washoff of detached sediment in tons/acre per interval
     DETS = detached sediment storage in tons/acre
```

WSSD is then subtracted from DETS.

Transport and detachment of soil particles from the soil matrix is simulated with the following equation:

```
     SCRSD = SURO/(SURS + SURO)*DELT60*KGER*((SURS + SURO)/DELT60)**JGER  (5)
where:
     SCRSD = scour of matrix soil in tons/acre per interval
     KGER  = coefficient for scour of the matrix soil
     JGER  = exponent for scour of the matrix soil
```

The sum of the two fluxes, WSSD and SCRSD, represents the total sediment outflow from the land segment.

The same algorithms are used for the simulation whether or not areal blocks are used. When blocks are used, block specific values for WSSD and SCRSD are calculated from the surface water fluxes and storages which correspond to each block.

Subroutine SOSED1 differs from SOSED2 in that it uses the dimensionally nonhomogeneous term (SURS + SURO)/DELT60 in the above equations, while SOSED2 uses the homogeneous term SURO/DELT60.

4.2(1).4.3 Remove by Surface Flow Using Method 2 (subroutine SOSED2)

Purpose

The purpose of this subroutine is the same as SOSED1. They only differ in method.

Method of Determining Removal

This method of determining sediment removal has not been tested. Unlike subroutine SOSED1, it makes use of the dimensionally homogeneous term SURO/DELT60 instead of (SURO+SURS)/DELT60.

The capacity of the overland flow to transport detached sediment is determined in this subroutine by:

STCAP = DELT60*KSER*(SURO/DELT60)**JSER (6)

When STCAP is more than the amount of detached sediment in storage, the flow washes off all of the detached sediment storage (DETS). However, when STCAP is less than the amount of detached sediment in storage, the situation is transport limiting, so WSSD is equal to STCAP.

Direct detachment and transport of the soil matrix by scouring (eg. gullying) is simulated with the equation:

SCRSD = DELT60*KGER*(SURO/DELT60)**JGER (7)

Definitions of the above terms can be found in subroutine SOSED2. The coefficents and exponents will have different values than in subroutine SOSED1 because they modify different variables.

4.2(1).4.4 Simulate Re-attachment of Detached Sediment (subroutine ATTACH)

Purpose

Subroutine ATTACH simulates the re-attachment of detached sediment (DETS) on the surface (soil compaction).

Method

Attachment to the soil matrix is simulated by merely reducing DETS. Since the soil matrix is considered to be unlimited, no addition to the soil matrix is necessary when this occurs. DETS is diminished at the start of each day that follows a day with no precipitation by multiplying it by (1.0 - AFFIX), where AFFIX is a parameter. This represents a first order rate of reduction of the detached soil storage.

4.2(1).5 Estimate Soil Temperatures (Section PSTEMP of Module PERLND)

Purpose

PSTEMP simulates soil temperatures for the surface, upper, and lower/groundwater layers of a land segment for use in module section PWTGAS and the agri-chemical sections. Good estimates of soil temperatures are particularly important for simulating first order transformations in the agri-chemical sections.

Method

The two methods used for estimating soil temperatures are based on the regression equation approach in the ARM Model (Donigian, et al. 1977) and the smoothing factor approach used in HSP QUALITY (Hydrocomp 1977) to simulate the temperatures of subsurface flows.

Simulation of soil temperatures is done by layers which correspond to those specified in the agri-chemical sections. The surface layer is the portion of the land segment that affects overland flow water quality characteristics. The subsurface layers are upper, lower, and groundwater. The upper layer affects interflow quality characteristics while the lower is a transition zone to groundwater. The temperature of the groundwater layer affects groundwater quality transformations and outflow characteristics. Lower layer and groundwater temperatures are considered approximately equal; a single value is estimated for both layers.

Surface layer soil temperatures are estimated by the following regression equation:

$$SLTMP = ASLT + BSLT*AIRTC \qquad (1)$$

where:

SLTMP = surface layer temperature in degrees C
ASLT = Y-intercept
BSLT = slope
AIRTC = air temperature in degrees C

Temperatures of the other layers are simulated by one of two methods. If TSOPFG is set equal to one in the User's Control Input, the upper layer soil temperature is estimated by a regression equation as a function of air temperature (similar to equation above), and the lower layer/groundwater layer temperature is specified by a parameter which can vary monthly. This method is similar to that used in the ARM Model except that ARM relates the upper layer temperature to the computed soil surface temperature instead of directly to air temperature.

If TSOPFG is set equal to zero, both the upper layer and the lower layer/groundwater layer temperatures are computed by using a mean departure from air temperature plus a smoothing factor. The same basic equation is used with separate state variables and parameters for each layer:

$$TMP = TMPS + SMO*(AIRTCS + TDIF - TMPS) \qquad (2)$$

where:

TMP = layer temperature at the end of the current interval in degrees C

SMO = smoothing factor (parameter)

AIRTCS = air temperature at the start of the current interval, Deg C

TDIF = parameter which specifies the difference between the mean air temperature and the mean temperature of the soil layer, Deg C

TMPS = layer temperature at the start of the current interval in degrees C

The values of the parameters for any of the layer computations can be linearly interpolated from monthly input values to obtain daily variations throughout the year. If this variation is not desired, the user may supply yearly values.

4.2(1).6 Estimate Water Temperature and Dissolved Gas Concentrations (Section PWTGAS of Module PERLND)

Purpose

PWTGAS estimates the water temperature and concentrations of dissolved oxygen and carbon dioxide in surface, interflow, and groundwater outflows from a pervious land segment.

Method

The temperature of each outflow is considered to be the same as the soil temperature of the layer from which the flow originates, except that water temperature can not be less than freezing. Soil temperatures must either be computed in module section PSTEMP or supplied directly as an input time series. The temperature of the surface outflow is equal to the surface layer soil temperature, the temperature of interflow to the upper layer soil temperature, and the temperature of the active groundwater outflow equals the lower layer and groundwater layer soil temperature.

The dissolved oxygen and carbon dioxide concentrations of the overland flow are assumed to be at saturation and are calculated as direct functions of water temperature. PWTGAS uses the following empirical nonlinear equation to relate dissolved oxygen at saturation to water temperature (Committee on Sanitary Engineering Research 1960):

$$SODOX = (14.652 + SOTMP*(-0.41022 + SOTMP*(0.007991 - 0.000077774*SOTMP)))*ELEVGC \qquad (1)$$

where:

SODOX = concentration of dissolved oxygen in surface outflow in mg/l
SOTMP = surface outflow temperature in degrees C
ELEVGC = correction factor for elevation above sea level (ELEVGC is calculated by the Run Interpreter dependent upon mean elevation of the segment)

The empirical equation for dissolved carbon dioxide concentration of the overland flow (Harnard and Davis 1943) is:

$$\mathrm{SOCO2} = (10{*}{*}(2385.73/\mathrm{ABSTMP} - 14.0184 + 0.0152642{*}\mathrm{ABSTMP})) {*}0.000316{*}\mathrm{ELEVGC}{*}12000.0 \qquad (2)$$

where:

SOCO2 = concentration of dissolved carbon dioxide in surface outflow in mg C/l
ABSTMP = absolute temperature of surface outflow in degrees K

The concentrations of dissolved oxygen and carbon dioxide in the interflow and the active groundwater flow cannot be assumed to be at saturation. Values for these concentrations are provided by the user. He may specify a constant value or 12 monthly values for the concentration of each of the gases in interflow and groundwater. If monthly values are provided, daily variation in values will automatically be obtained by linear interpolation between the monthly values.

4.2(1).7 Simulate Quality Constituents Using Simple Relationships with Sediment and Water Yield (Section PQUAL of Module PERLND)

Purpose

The PQUAL module section simulates water quality constituents or pollutants in the outflows from a pervious land segment using simple relationships with water and/or sediment yield. Any constituent can be simulated by this module section. The user supplies the name, units and parameter values appropriate to each of the constituents that he wishes to simulate. However, more detailed methods of simulating sediment, heat, dissolved oxygen, dissolved carbon dioxide, nitrogen, phosphorus, soluble tracers, and pesticide removal from a pervious land segment are available, in other module sections.

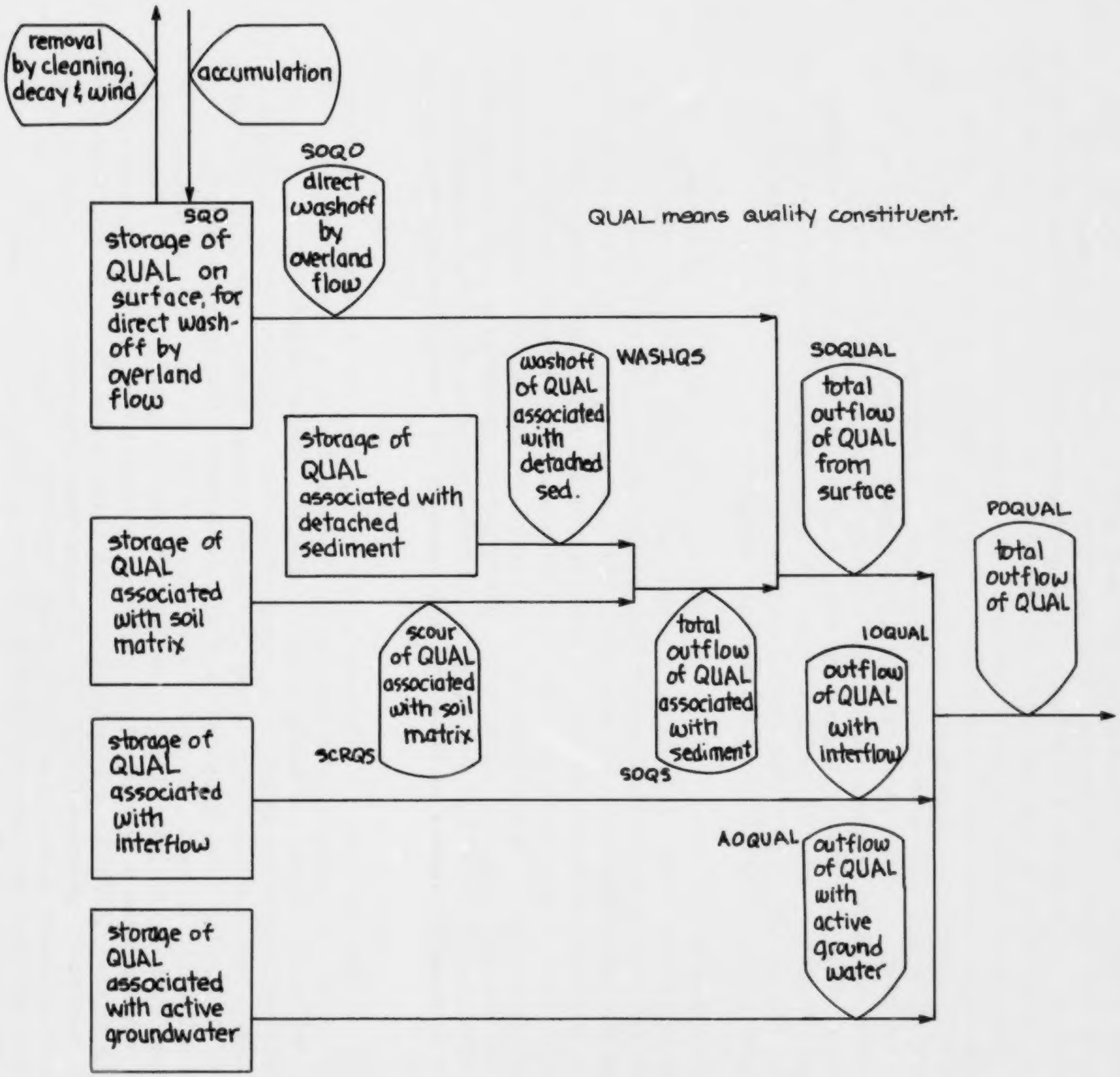
Approach

The basic algorithms used to simulate quality constituents are a synthesis of those used in the NPS Model (Donigian and Crawford 1976b) and HSP QUALITY (Hydrocomp 1977). However, some options and combinations are unique to HSPF.

Figure 4.2(1).7-1 shows schematically the fluxes and storages represented in module section PQUAL. The occurrence of a water quality constituent in both surface and subsurface outflow can be simulated. The behavior of a constituent in surface outflow is considered more complex and dynamic than the behavior in subsurface flow. A constituent on the surface can be affected greatly by adhesion to the soil and by temperature, light, wind, and direct human influences. Section PQUAL is able to represent these processes in a general fashion. It allows quantities in the surface outflow to be simulated by two methods. One approach is to simulate the constituent by association with sediment removal. The other approach is to simulate it using basic accumulation and depletion rates together with depletion by washoff; that is, constituent outflow from the surface is a function of the water flow and the constituent in storage. A combination of the two methods may be used in which the individual outfluxes are added to obtain the total surface outflow. These approaches will be discussed further in the descriptions of the corresponding subroutines. Concentrations of quality constituents in the subsurface flows of interflow and active groundwater are supplied by the user. The concentration may be linearly interpolated to obtain daily values from input monthly values.

The user has the useful option of simulating the constituents by any combination of these surface and subsurface outflow pathways. The outflux from the combination of the pathways simulated will be the total outflow from the land segment. In addition, the user is able to select the units to be associated with the fluxes. These options give the user considerable flexibility. For example, he may wish to simulate coliforms in units of organisms/acre by association with sediment in the surface runoff and using a concentration in the groundwater which varies seasonally. Or he may want to simulate total dissolved salts in pounds per acre by direct association with overland flow and a constant concentration in interflow and groundwater flow.

PQUAL allows the user to simulate up to 10 quality constituents at a time. Each of the 10 constituents may be defined as one or a combination of the following types: QUALSD, QUALOF, QUALIF, and/or QUALGW. If a constituent is considered to be associated with sediment, it is called a QUALSD. The corresponding terms for constituents associated with overland flow, interflow, and groundwater flow are QUALOF, QUALIF, and QUALGW, respectively. However, no more than seven of any one of the constituent types (QUALSD, QUALOF, QUALIF, or QUALGW) may be simulated in one operation. The program uses a set of flag pointers to keep track of these associations. For example, QSDFP(3) = 0 means that the third constituent is not associated with sediment, whereas QSDFP(6) = 4 means that the sixth constituent is the fourth sediment associated constituent (QUALSD). Similar flag pointer arrays are used to indicate whether or not a quality constituent is a QUALOF, QUALIF, or QUALGW.


removal by cleaning, decay & wind
accumulation
SQO
storage of QUAL on surface, for direct wash-off by overland flow
SOQO
direct washoff by overland flow
storage of QUAL associated with detached sediment
washoff of QUAL associated with detached sed.
WASHQS
storage of QUAL associated with soil matrix
scour of QUAL associated with soil matrix
SCRQS
total outflow of QUAL associated with sediment
SOQS
SOQUAL
total outflow of QUAL from surface
POQUAL
total outflow of QUAL
IOQUAL
outflow of QUAL with interflow
storage of QUAL associated with interflow
AOQUAL
outflow of QUAL with active ground water
storage of QUAL associated with active groundwater

QUAL means quality constituent.

Figure 4.2(1).7-1 Flow diagram for PQUAL section of PERLND Application Module.

4.2(1).7.1 Remove by Association with Sediment (subroutine QUALSD)

Purpose

QUALSD simulates the removal of a quality constituent from a pervious land surface by association with the sediment removal determined in module section SEDMNT.

Method

This approach assumes that the particular quality constituent removed from the land surface is in proportion to the sediment removal. The relation is specified with user-input "potency factors." Potency factors indicate the constituent strength relative to the sediment removed from the surface. Various quality constituents such as iron, lead, and strongly adsorbed toxicants are actually attached to the sediment being removed from the land surface. Some other pollutants such as ammonia, organics, pathogens, and BOD may not be extensively adsorbed, but can be considered highly correlated to sediment yield.

For each quality constituent associated with sediment, the user supplies separate potency factors for association with washed off and scoured sediment (WSSD and SCRSD). Typically, the washoff potency factor would be larger than the scour potency factor because washed off sediment is usually finer than the scoured material and thus has a higher adsorption capacity. Organic nitrogen would be a common example of such a constituent. The user is also able to supply monthly potency factors for constituents that vary somewhat consistently during the year. For instance, constituents that are associated with spring and fall fertilization may require such monthly input values.

Removal of the sediment associated constituent by detached sediment washoff is simulated by:

$$WASHQS = WSSD*POTFW \qquad (1)$$

where:

WASHQS = flux of quality constituent associated with detached sediment washoff in quantity/acre per interval
WSSD = washoff of detached sediment in tons/acre per interval
POTFW = washoff potency factor in quantity/ton

Removal of constituents by scouring of the soil matrix is similar:

$$SCRQS = SCRSD*POTFS \qquad (2)$$

where:

SCRQS = flux of quality constituent associated with scouring of the matrix soil in quantity/acre per interval

SCRSD = scour of matrix soil in tons/acre per interval
POTFS = scour potency factor in quantity/ton

WASHQS and SCRQS are combined to give the total sediment associated flux of the constituent from the land segment, SOQS.

The unit "quantity" refers to mass units (pounds or tons in the English system) or some other quantity, such as number of organisms for coliforms. The unit is user specified.

4.2(1).7.2 Accumulate and Remove by a Constant Unit Rate and by Overland Flow (subroutine QUALOF)

Purpose

QUALOF simulates the accumulation of a quality constituent on the pervious land surface and its removal by a constant unit rate and by overland flow.

Method

This subroutine differs from the others in module section PQUAL in that the storage of the quality constituent on the land surface is simulated. The constituent can be accumulated and removed by processes which are independent of storm events such as cleaning, decay, and wind erosion and deposition, or it can be washed off by overland flow. The accumulation and removal rates can have monthly values to account for seasonal fluctuations. A pollution indicator such as fecal coliform from range land is an example of a constituent with accumulation and removal rates which may need to vary throughout the year. The concentration of the coliform in the surface runoff may fluctuate with the seasonal grazing density, and the weather.

When there is surface outflow and some quality constituent is in storage, washoff is simulated using the commonly used relationship:

$$SOQO = SQO*(1.0 - EXP(-SURO*WSFAC)) \qquad (3)$$

where:

SOQO = washoff of the quality constituent from the land surface in quantity/acre per interval
SQO = storage of available quality constituent on the surface in quantity/acre
SURO = surface outflow of water in in./interval
WSFAC = susceptibility of the quality constituent to washoff in units of 1/in.
EXP = Fortran exponential function

The storage is updated once a day to account for accumulation and removal which occurs independent of runoff by the equation:

$$\mathrm{SQO} = \mathrm{ACQOP} + \mathrm{SQOS} * (1.0 - \mathrm{REMQOP}) \qquad (4)$$

where:

ACQOP = accumulation rate of the constituent, quantity/acre per day
SQOS = SQO at the start of the interval
REMQOP = unit removal rate of the stored constituent, per day

The Run Interpreter computes REMQOP and WSFAC for this subroutine according to:

$$\mathrm{REMQOP} = \mathrm{ACQOP}/\mathrm{SQOLIM} \qquad (5)$$

where:

SQOLIM = asymptotic limit for SQO as time approaches infinity (quantity/acre), if no washoff occurs

and

$$\mathrm{WSFAC} = 2.30/\mathrm{WSQOP} \qquad (6)$$

where:

WSQOP = rate of surface runoff which results in a 90 percent washoff in one hour, in./hr

Since the unit removal rate of the stored constituent (REMQOP) is computed from two other parameters, it does not have to be supplied by the user.

4.2(1).7.3 Simulate by Association with Interflow Outflow (subroutine QUALIF)

Purpose

QUALIF is designed to permit the user to simulate the occurrence of a constituent in interflow.

Method

The user simply specifies a concentration for each constituent which is a QUALIF. An option permits him to supply 12 monthly values, to account for seasonal fluctuations. In this case, the system interpolates a new value each day.

4.2(1).7.4 Simulate by Association with Active Goundwater Outflow (subroutine QUALGW)

Purpose

QUALGW is designed to permit the user to simulate the occurrence of a constituent in ground water outflow.

Method

Identical to that for QUALIF

Introduction to the Agri-chemical Sections

The introduction of agricultural chemicals into streams, lakes, and groundwater from agricultural land may be detrimental. For example, persistant fat soluble pesticides, such as DDT, have been known to concentrate in the fatty tissue of animals causing toxic effects. Nitrogen and phosphorus are essential plant nutrients which when introduced into certain surface waters will increase productivity. This may or may not be desirable depending upon management objectives. Significant productivity results in algal blooms, but some increase in productivity will increase fish production. Drinking water containing high nitrate concentrations may cause methomoglobinemia in small children.

Pesticide, nitrogen, and phosphorus compounds are important to agricultural production, but prediction of their removal from the field is necessary for wise management of both land and water resources. HSPF can be used to predict such outflows. The agri-chemical sections of the PERLND module of HSPF simulate in detail nutrient and pesticide processes, both biological and chemical, and the movement of any nonreactive tracer in a land segment. These chemicals can also be simulated in module section PQUAL but in a simplified manner. The dynamic and continuous processes that affect the storages and outflow of pesticides and of nutrients from fertilized fields should be simulated in detail to fully analyze agricultural runoff. If the situation does not require full representation of these processes, or if data are not available, the PQUAL subroutines could be used.

The basic algorithms in the agri-chemical sections of HSPF were originally developed for use on agricultural lands, but can be used on other pervious areas where pesticides and plant nutrients occur, for example, orchards, nursery land, parks, golf courses, and forests. All pervious land contains nitrogen and phosphorus in the soil; it is possible to use this module to simulate the behavior of agricultural chemicals in any such area.

Comparison of HSPF and ARM

The methods used to simulate pesticide processes in the agri-chemical sections were developed orginally for the Pesticide Transport and Runoff (PTR) Model (Crawford and Donigian 1973), then expanded to include nutrients in the Agricultural Runoff Management (ARM) Model (Donigian and Crawford 1976) and tested and modified in ARM Version II (Donigian, et al. 1977). In HSPF the ARM Version II algorithms were recreated with some additional options. (For more detail on the basic methods, refer to the above reports.)

The differences between HSPF and ARM Model Version II should, however, be discussed. The biggest difference is the availability of new options to simulate soil nutrient and pesticide adsorption and desorption. Ammonium and phosphate adsorption/desorption in HSPF can be accomplished by using Freundlich isotherms as well as by first order kinetics. Pesticides can be adsorbed and desorbed by the two Freundlich methods used in the ARM Model or by first order kinetics. In addition, the pesticide parameter values are now input for each separate soil layer instead of inputting one parameter set for all the layers. HSPF also allows the user to simulate more than one pesticide in a run. (The ARM Model only simulates one per run.) In addition to the percolation factors which can still be used to retard any solute leaching from the upper layer and lower layer, a multiplication factor has been introduced that can reduce leaching from the surface layer. Also, in HSPF, nitrogen and phosphorus chemical and biochemical transformations can each be simulated at different time steps to save computer time. Plant uptake of ammonium is another new option in HSPF.

The topsoil layers can still be divided into areal blocks (see the PWATER section). They have been used in the ARM Model to represent areal variation in chemical concentrations over the land surface. They divide the hyrologic responses and chemical storages in the surface and upper layers into conceptual areal zones based on hydrologic variations. HSPF allows the user to use the model with from one to five blocks (Section 4.2(1).3); whereas the ARM Model uses a fixed five block setup. The capability to vary the number of blocks allows experimentation. In some situations a land segment may not need to be subdivided into more than one block, resulting in substantial savings in computer costs.

Units

The fluxes and storages of chemicals modeled in these module sections are in mass per area units. The user must supply his input in appropriate units; kg/ha if he is using the Metric system, and lb/ac for the English system. Internally, most of the code does not differentiate between the unit systems. Fluxes are determined by either proportionality constants, fractions of chemicals in storage, or unitless concentrations. First order kinetics makes use of proportionality constants for determining reaction fluxes. Chemicals are transported based on the fractions of that in storage. Freundlich adsorption/desorption is based on ppm concentrations.

Module Sections

There are five agri-chemical module sections. They are shown in the structure chart of PERLND (No. 4.2(1)). Module section MSTLAY manipulates water storages and fluxes calculated in module section PWATER. This section must be run before the following sections can be run, since it supplies them with data for simulating the storage and movement of solutes. Module section PEST simulates pesticide behavior while NITR and PHOS simulate the plant nutrients of nitrogen and phosphorus. Simulation of a nonreactive solute (tracer) is accomplished in module section TRACER.

4.2(1).8 Estimate Moisture Content of Soil Layers and Fractional Fluxes (Section MSTLAY of Module PERLND)

Purpose

This module section estimates the storages of moisture in the four soil layers with which the agricultural chemical sections deal (Figure 4.2(1).8-1); and the fluxes of moisture between the storages. MSTLAY is required because the moisture storages and fluxes computed by module section PWATER can not be directly used to simulate solute transport through the soil. For example, in PWATER, some moisture which infiltrates can reach the ground water in a single time step (Figure 4.2(1).3-2). While this phenomenon does not have any serious effect in simulating the hydrologic response of a land segment, it does seriously affect the simulation of solute transport.

Thus, MSTLAY takes the fluxes and storages computed in PWATER and adapts them to fit the storage/flow path picture in Figure 4.2(1).8-1. The revised storages, in inches of water, are also expressed in mass/area units (that is, lb/acre or kg/ha) for use in the adsorption/desorption calculations.

Method

Figure 4.2(1).8-1 schematically diagrams the moisture storages and fluxes used in subroutine MSTLAY. Note that the fluxes are represented in terms of both quantity (eg. IFWI, in inches/interval) and as a fraction of the contributing storage (eg. FII, as a fraction of UMST/interval)

The reader should also refer to Figure 4.2(1).3-2 in module section PWATER when studying this diagram and the following discussion.

For the agri-chemical sections the moisture storages (the variables in Figure 4.2(1).8-1 ending in MST) are calculated by the general equation:

$$\text{MST} = \text{WSTOR} + \text{WFLUX} \qquad (1)$$

The variable WSTOR is the related storage calculated in module section PWATER (Figure 4.2(1).3-2). For example, in the calculation of the lower layer moisture storage (LMST), WSTOR is the lower zone storage (LZS). The variable WFLUX generally corresponds to the flux of moisture through the soil layer. For the computation of LMST, WFLUX is the sum of water percolating from the lower zone to the inactive (IGWI) and active groundwater (AGWI) as determined in section PWATER. Note that these equations are dimensionally non-homogeneous, because storages (inches) and fluxes (inches/interval) are added together. Thus, the results given are likely to be highly dependent on the simulation time step. The ARM Model, from which the equations come, uses a step of 5 minutes. Extreme caution should be exercised if the agricultural chemical sections (including MSTLAY) are run with any other time step. For more details on the calculation of the layer moisture storages, the reader should consult the pseudo code.

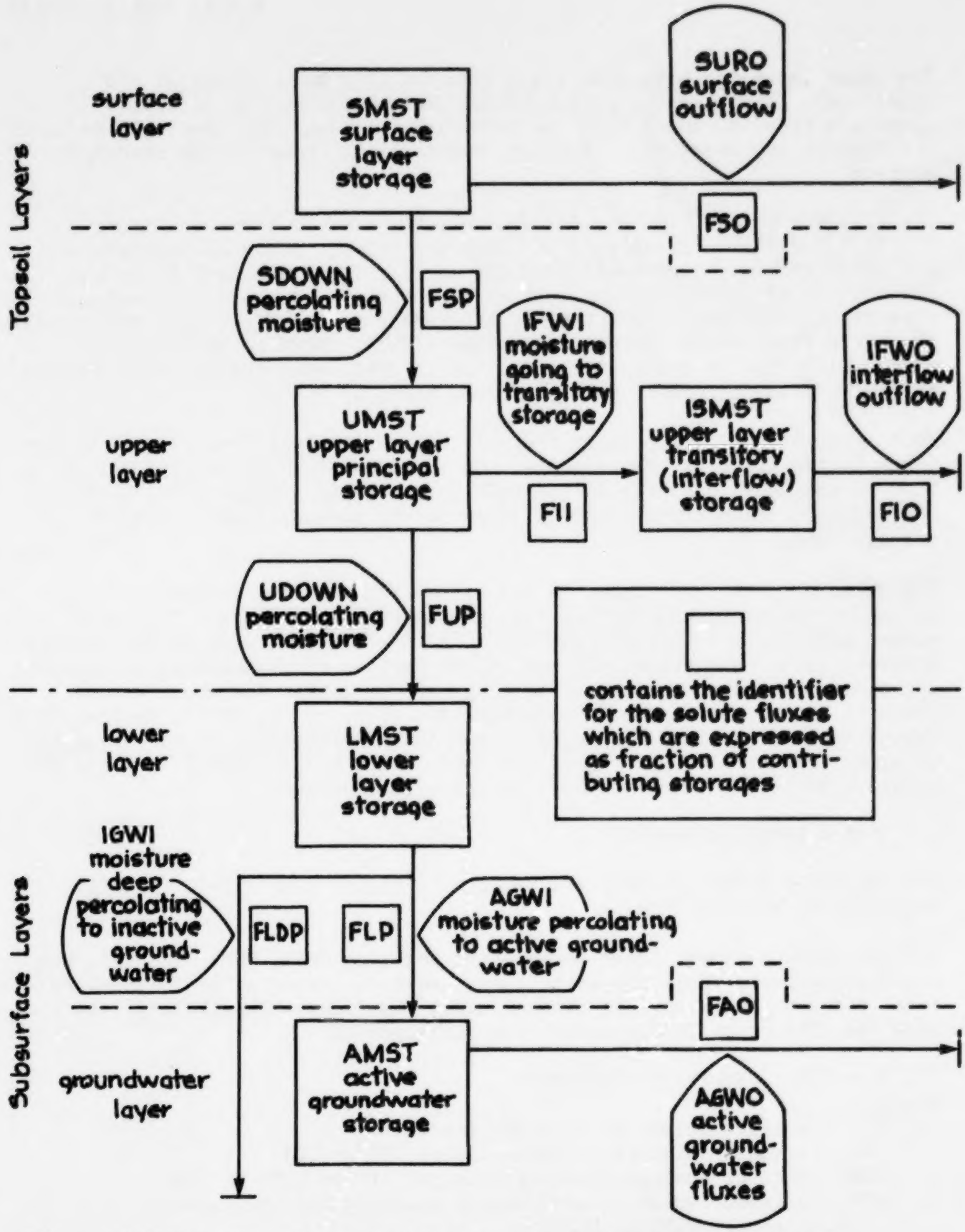

Topsoil Layers
Subsurface Layers
surface layer
upper layer
lower layer
groundwater layer
SURO surface outflow
SMST surface layer storage
FSO
SDOWN percolating moisture
FSP
IFWI moisture going to transitory storage
IFWO interflow outflow
UMST upper layer principal storage
ISMST upper layer transitory (interflow) storage
FII
FIO
UDOWN percolating moisture
FUP
contains the identifier for the solute fluxes which are expressed as fraction of contributing storages
LMST lower layer storage
IGWI moisture deep percolating to inactive groundwater
FLDP
FLP
AGWI moisture percolating to active groundwater
FAO
AMST active groundwater storage
AGWO active groundwater fluxes

Figure 4.2(1).8-1 Flow diagram of the transport of moisture & solutes as estimated in the MSTLAY section of the PERLND Application Module.

The upper layer has been subdivided into two storages, principal and transitory. The transitory (interflow) storage is used to transport chemicals from the upper layer to interflow outflow. The chemicals in it do not undergo any reactions. However, reactions do occur in the principal storage.

The fluxes shown in Figure 4.2(1).8-1 are the same as those in Figure 4.2(1).3-2 with the exceptions of SDOWN and UDOWN. SDOWN encompasses all the water that moves downward from the surface layer storage. It is the combination of the water infiltrating from the surface detention storage directly to the lower zone (INFIL), the inflow to the upper zone (UZI), and the water flowing into interflow storage (IFWI). UDOWN is all the water percolating through the upper layer. It is INFIL plus the percolation from the upper zone storage to the lower zone storage (PERC).

Each fractional solute flux is the appropriate moisture flux divided by the contributing storage. For example, the fraction of chemical in solution that is transported overland from the surface layer storage (FSO) is the surface moisture outflow (SURO) divided by the surface layer moisture storage (SMST).

The above estimates are based on the assumption that the concentration of the solute being transported is the same as that in storage. They also assume uniform flow through the layers and continuous mixing of the solutes. However, these assumptions may need to be revised or implemented differently for some of the transport. Past testing has shown that the above method leads to excessive leaching of solutes (Donigian, et al. 1977). Factors that retard solute leaching were added in the ARM Model Version II to remedy this problem. For the surface layer, the percolation factor (SLMPF) affects the solute fraction percolating (FSP) by the relationship:

$$FSP = SLMPF*SDOWN/SMST \qquad (2)$$

The variables SDOWN and SMST are defined in Figure 4.2(1).8-1. FSP will typically be between 0 and 1.

For the upper or lower layer percolating fraction (FUP, FLDP, or FLP), the retardation factor only has an influence when the ratio of the respective zonal storage to the nominal storage times the factor (ZS/(ZSN*LPF)) is less than one. The relationship under this condition is:

$$F = (ZS/(ZSN*LPF))*(PFLUX/MST) \qquad (3)$$

where:

F = layer solute percolating fraction
ZS = zonal moisture storage, either UZS or LZS
ZSN = zonal nominal moisture storage, either UZSN or LZSN
LPF = factor which retards solute leaching for the layer, either ULPF or LLPF
PFLUX = percolation flux, either UDOWN, IGWI, or AGWI
MST = layer moisture storage, either UMST or LMST

4.2(1).9 Simulate Pesticide Behavior in Detail (Section PEST of Module PERLND)

Purpose

Because of the complexity of pesticide behavior on the land, simulation of the processes frequently requires considerable detail. Pesticide applications vary in amount and time during the year. Various pesticides adsorb and degrade differently. Some, like paraquat, attach themselves strongly to the soil thereby appearing in low concentrations in water but in high concentrations on soil particles. Others, like atrazine, undergo complex interactions with the soil and are found in higher concentrations in the runoff water than on the eroded sediment.

Section PEST models pesticide behavior by simulating the processes of degradation and adsorption as well as transport. The pesticides are simulated in the soil and runoff in three forms: dissolved, adsorbed, and crystallized. These phases in the soil affect the forms and amounts in the runoff.

Method

Pesticides are simulated by using the time series generated by other PERLND module sections to transport and influence the adsorption and degradation processes. Pesticides move with water flow or by association with the sediment. They also may be adsorbed to the soil in varing degree as a function of the chemical characteristics of the toxicant and the exchange capacity of the soil layer. Pesticide degradation occurs to varying degrees depending upon the susceptability of the compound to volatilization and breakdown by light, heat, microorganisms and chemical processes. The subroutines in module section PEST consider these transport and reaction processes.

All the subroutines described in this module section except NONSV and DEGRAS are accessed by other agri-chemical module sections because many of the basic transport and reaction processes are similar. The subroutines are described here because they are physically located in this subroutine group. Subroutine AGRGET is first to be called. This subroutine has no computing function; it obtains any required time series from the INPAD that is not already available.

Subroutine SDFRAC determines the fraction of the surface layer soil that has eroded. The amount eroded is the total sediment removed by scour and washoff as determined in module section SEDMNT. The mass of soil in the surface layer is a parameter value which does not vary even when material is removed. The chemical which is associated with the sediment is assumed to be removed from the surface layer storage in the same proportion that the layer has eroded. Chemical removal is simulated in subroutine SEDMOV. A sediment associated chemical is one that may be attached to the eroding soil or one which may move with the soil. With pesticides the adsorbed form will be attached to the soil particle, while the crystalline form will move with the soil particle being eroded but will not be attached to it. Both forms are taken from their respective surface layer storages in proportion to the fraction of the surface soil layer removed by overland flow.

Subroutines TOPMOV and SUBMOV perform a function similar to SEDMOV except they move the solutes. Chemicals in solution move to and from the storages according to the fractions calculated in section MSTLAY. Figure 4.2(1).9-1 schematically illustrates the fluxes and storages used in these subroutines. The fractions (variables beginning with the letter "F") of the storages are used to compute the solute fluxes. The equations used to compute the solute transport fluxes from the fractions and storages are given in the figure. Subroutine TOPMOV performs the calculations of the fluxes and the resulting changes in storage for the topsoil layers (surface and upper), while SUBMOV performs them for the subsurface layers (lower and active groundwater).

Biological and chemical reactions are performed on the pesticides (and other chemicals) in each layer storage. Chemicals in the upper layer principal storage undergo reactions while those in the transitory (interflow) storage do not. The upper layer transitory storage is a temporary storage of chemicals on their way to interflow outflow. Subroutine PSTRXN is called to perform reactions on the pesticide in each layer.

4.2(1).9.5 Perform Reactions on Pesticides (subroutine PSTRXN)

Purpose

This code simulates the degradation and adsorbtion/desorption of pesticides. This subroutine is called for each of the four soil layers and each pesticide.

Method of Reacting Pesticides

The user has the option of adsorbing/desorbing the pesticide by one of three methods. The first method is by first order kinetics. This method assumes that the pesticide adsorbs and desorbs at a rate based on the amount in soil solution and on the amount on the soil particle. It makes use of a proportionality constant and is independent of the concentration. The second method is by use of the single value Freundlich isotherm. This method makes use of a single adsorption/desorption curve for determining the concentration on the soil and in solution. The third method is by use of multiple curves based on a varying Freundlich K value. Further details of these methods can be found in the discussion of the individual subroutines that follows and in the ARM Model reports (Donigian and Crawford 1976; Donigian, et al. 1977).

Degradation is performed once a day by subroutine DEGRAS for each of the four layers that contain pesticide. The amount degraded is determined simply by multiplying a decay rate parameter specified for each soil layer by each of the three forms (adsorbed, solution, and crystalline) of pesticide in storage. The degraded amounts are then subtracted from their respective storages. This method of simulating degradation lumps complex processes in a simple parameter.

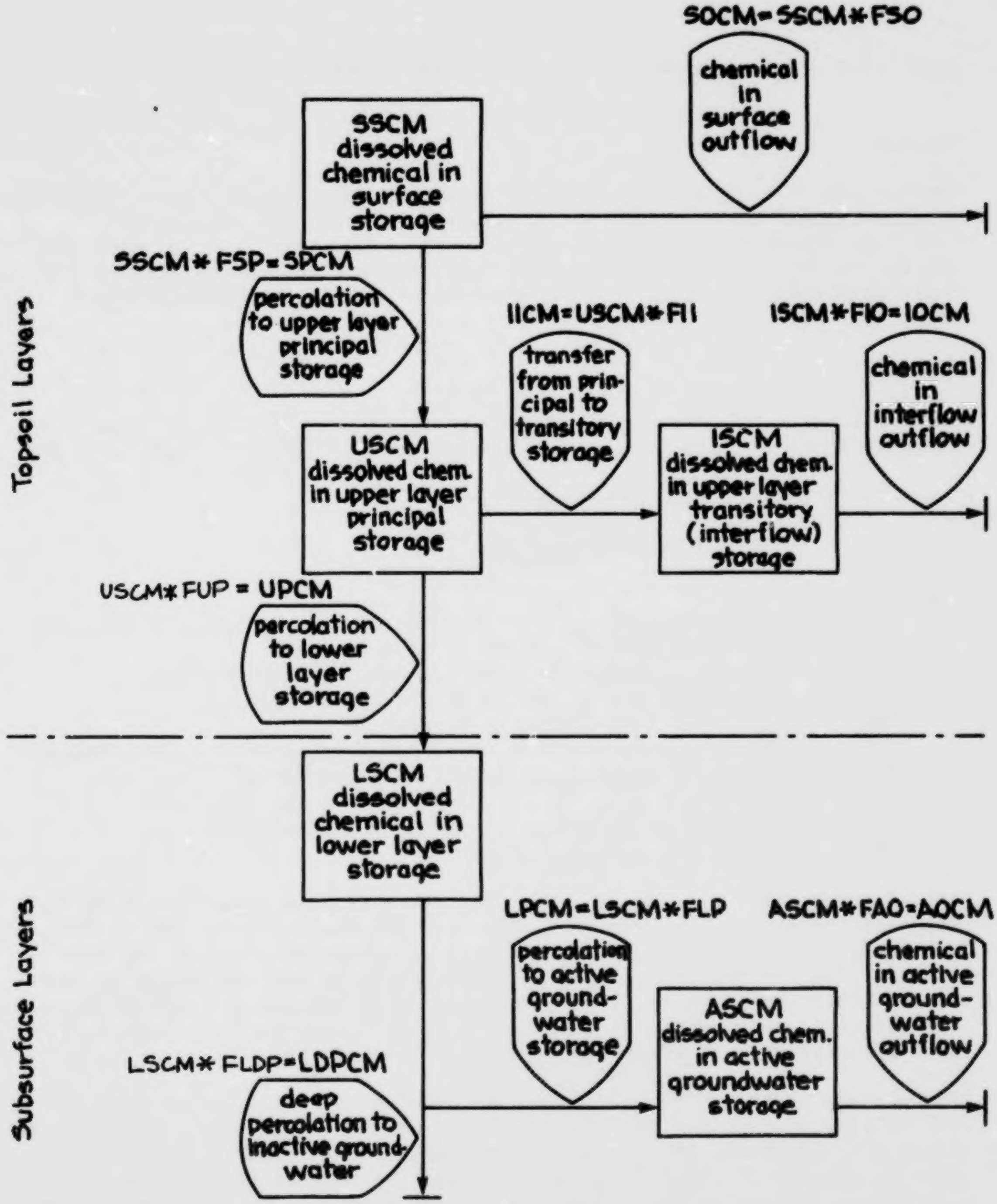

SOCM=SSCM*FSO
chemical in surface outflow
SSCM dissolved chemical in surface storage
SSCM*FSP=SPCM
percolation to upper layer principal storage
IICM=USCM*FII
transfer from principal to transitory storage
ISCM*FIO=IOCM
chemical in interflow outflow
USCM dissolved chem. in upper layer principal storage
ISCM dissolved chem. in upper layer transitory (interflow) storage
USCM*FUP = UPCM
percolation to lower layer storage
Topsoil Layers
LSCM dissolved chemical in lower layer storage
LPCM=LSCM*FLP
percolation to active ground-water storage
ASCM*FAO=AOCM
chemical in active ground-water outflow
ASCM dissolved chem. in active groundwater storage
LSCM*FLDP=LDPCM
deep percolation to inactive ground-water
Subsurface Layers

Figure 4.2(1).9-1 Flow diagram showing modeled movement of chemicals in solution

4.2(1).9.5.1 Adsorb/Desorb Using First Order Kinetics (subroutine FIRORD)

Purpose

The purpose of this subroutine is to calculate the adsorption and desorption reaction fluxes of chemicals using temperature dependent first order kinetics. These fluxes are calculated every simulation interval when the subroutine is called by section PEST, but they are determined only at the designated chemical reaction frequency when called by sections NITR and PHOS.

Method

The calculation of adsorption and desorption reaction fluxes by first order kinetics for soil layer temperatures less than 35 degrees C takes the form:

```
DES = CMAD*KDS*THKDS**(TMP-35.0)                                (1)

ADS = CMSU*KAD*THKAD**(TMP-35.0)                                (2)
```

where:
DES = current desorption flux of chemical in mass/area per interval
CMAD = storage of adsorbed chemical in mass/area
KDS = first order desorption rate parameter, per interval
THKDS = temperature correction parameter for desorption
TMP = soil layer temperature in degrees C
ADS = current adsorption flux of chemical in mass/area per interval
CMSU = storage of chemical in solution in mass/area
KAD = first order adsorption rate parameter, per interval
THKAD = temperature correction parameter for adsorption
THKDS and THKAD are typically about 1.06

All of the variables except the temperature coefficients may vary with the layer of the soil being simulated. The soil temperatures are time series which may be input (eg. using field data) or simulated in module section PSTEMP. The temperature correction of the reaction rate parameter is based on the Arrhenius equation. At temperatures of 35 degrees C or above no correction is made. When the temperature is at 0 degrees C or below or the soil layer is dry, no adsorption and desorption occurs.

The storage of the solution chemical is updated every simulation interval in the calling subroutine, that is, in PSTRXN, NITRXN, or PHORXN, by adding DES minus ADS. Likewise, the storage of the adsorbed chemical is updated there also by adding ADS minus DES. An adjustment is made in the calling subroutine, if any of the fluxes would cause a storage to go negative. When this happens a warning message is produced and fluxes are adjusted so that no storage goes megative. This usually occurs when large time steps are used in conjuction with large KAD and KDS values.

4.2(1).9.5.2 Adsorb/Desorb Using the Single Value Freundlich Method (subroutine SV)

Purpose

Subroutine SV calculates the adsorption and desorption and the resulting new storages of a chemical using the single value Freundlich method.

Method

The Freundlich isotherm methods, unlike first order kinetics, assume instantaneous equilibrium. That is, no matter how much chemical is added to a particular phase, equilibrium is assumed to be established between the solution and adsorbed phase of the chemical. These methods also assume that for any given amount of chemical in the soil, the equilibrium distribution of the chemical between the soil solution and on the soil particle can be found from an isotherm. Figure 4.2(1).9-2 illustrates such an isotherm.

Three phases of the chemical are actually possible; crystalline, adsorbed, and solution. The crystalline form is assumed to occur only when the soil layer is dry, or when there is more chemical in the layer than the combined capacity to adsorb and hold in solution. When the soil is dry, all the chemical is considered to be crystalline salt. When there is more total chemical in the soil layer than the the soil adsoption sites can contain and more than that saturated in solution, then the chemical content which exceeds these capacities is considered to be crystalline salt. Module section PEST considers crystalline phase storage, but in module sections NITR and PHOS this is not so. Instead, any crystalline phosphate or ammonium predicted by an isotherm is added to the adsorbed phase storage.

The adsorbed and solution phases of the chemical are determined in this subroutine by the standard Freundlich equation as plotted by curve 1 in Figure 4.2(1).9-2. When the amount of chemical is less than the capacity of the soil particle lattice to permanently bind the chemical (XFIX), then all the material is consider fixed. All the fixed chemical is contained in the adsorbed phase of the layer storage. Otherwise, the Freundlich equation for curve 1 is used to determine the partitioning of the chemical into the adsorbed and solution phases:

`X = KF1*C**(1/N1) + XFIX` (3)

where:

X = chemical adsorbed on soil, in ppm of soil
KF1 = single value Freundlich K coefficient
C = equilibrium chemical concentration in solution, in ppm of solution
N1 = single value Freundlich exponent
XFIX = chemical which is permanently fixed, in ppm of soil

The above equation is solved in subroutine ITER by an iteration technique. The parameters used in the computation can differ for each layer of the soil.

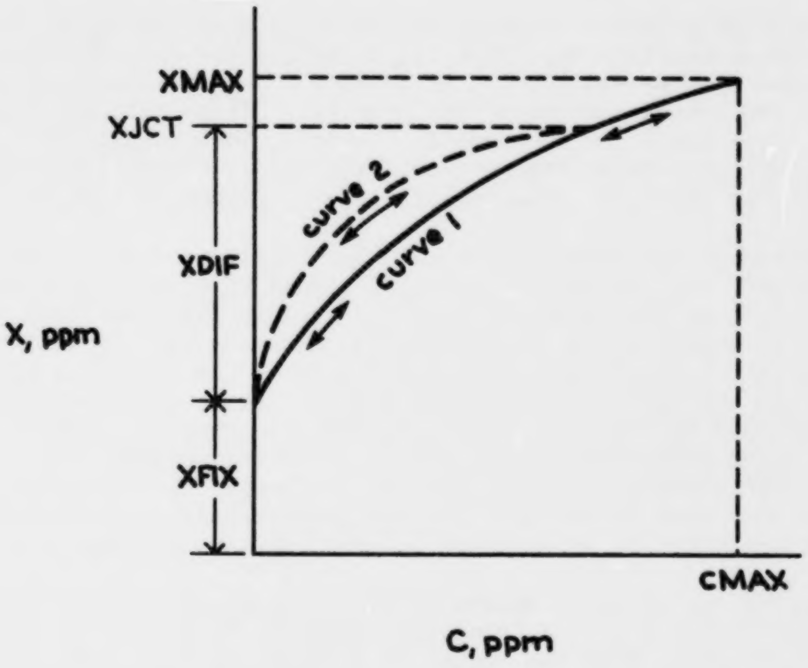


Figure 4.2(1).9-2 Freundlich Isotherm Calculations. X is the chemical adsorbed on the soil and C is the chemical in solution.

4.2(1).9.5.3 Adsorb/Desorb Using the Non-single Value Freundlich Method (subroutine NONSV)

Purpose

The purpose of this subroutine is to calculate the adsorption/desorption of a chemical by the nonsingle value Freundlich method. The single value Freundlich method was found to inadequately represent the division of some pesticides between the soil particle and solution phases, so this method was developed as an option in the ARM Model (Donigian and Crawford 1976). This subroutine is only available for use by the PEST module section.

Method

The approach in this code uses the same algorithms and solution technique as subroutine SV for determining curve 1 in Figure 4.2(1).9-2. However, curve 1 is used solely for adsorption. That is, only when the concentration of the adsorbed chemical is increasing. When desorption occurs a new curve (curve 2) is used:

```
X = KF2*C**(1/N2) + XFIX                                       (4)

KF2 = (KF1/XDIF)**(N1/N2) * XDIF                               (5)
```

where:

KF2 = nonsingle value Freundlich coefficient
N2   = nonsingle value Freundlich exponent parameter
XDIF = XJCT - XFIX
XJCT = the adsorbed concentration where curve 1 joins curve 2 (ie. where desorption started) as shown in Figure 4.2(1).9-2, in ppm of soil

The other variables are as defined for subroutine SV.

Once curve 2 is used, both desorption and adsorption follow it until the adsorbed concentration is less than or equal to XFIX or until it reaches XJCT. Then, adsorption will again take place following curve 1 until desorption reoccurs, following a newly calculated curve 2. The solution of the Freundlich equations for curves 1 and 2 utilizes the same iteration technique introduced in subroutine SV (subroutine ITER).

4.2(1).10 Simulate Nitrogen Behavior in Detail (section NITR of module PERLND)

Purpose

NITR, like section PEST, simulates the behavior of chemicals in the soil profile of a land segment. Section NITR handles the nitrogen species of nitrate, ammonia, and organic nitrogen. This involves simulating nitrogen transport and soil reactions. Nitrogen, like phosphorus, may be a limiting nutrient in the eutrophication process in lakes and streams. Nitrates in high concentrations may also pose a health hazard to infants.

Method of Simulating Nitrogen

Nitrogen species are transported by the same methods used for the pesticide forms. The subroutines that are called to transport nitrogen are located in and described with the PEST module section. Organic nitrogen and adsorbed ammonium are removed from the surface layer storage by association with sediment (subroutines SDFRAC and SEDMOV). Nitrate and ammonium in the soil water are transported using the simulation subroutines TOPMOV and SUBMOV.

Nitrogen reactions are simulated separately for each of the soil layers. The method is discussed below.

4.2(1).10.1 Perform Reactions on Nitrogen Forms (subroutine NITRXN)

Purpose

The purpose of NITRXN is to simulate soil nitrogen transformations. This includes plant uptake of nitrate and ammonium, denitrification or reduction of nitrate-nitrite, immobilization of nitrate-nitrite and ammonium, mineralization of organic nitrogen, and the adsorption/desorption of ammonium.

Method of Nitrogen Transformations

Nitrogen reactions can be divided between those that are chemical in nature and those that are a combination of chemical and biological reactions. The adsorption and desorption of ammomium is a chemical process. The user has the option of simulating ammonium adsorption and desorption by first order kinetics with subroutine FIRORD or by the Freundlich isotherm method with subroutine SV. These subroutines are described in the PEST module section.

The user has the option of specifying how often the adsorption and desorption rates are calculated. When adsorption/desorption is simulated by the Freundlich method, the solution and adsorbed storages of ammonium are determined instantaneously at the specified frequency of reaction. However, when the first order method is used, the temperature corrected reaction fluxes (Figure 4.2(1).10-1) are recomputed intermittently, but the storages are updated every simulation interval.

The other reactions are a combination of biological and chemical transformations. They are accomplished by first order kinetics only. The optimum first order kinetic rate parameter is corrected for soil temperatures below 35 degrees C by the generalized equation:

$$KK = K*TH**(TMP-35.0) \tag{1}$$

where:

KK = temperature corrected first order transformation rate in units of per simulation interval
K = optimum first order reaction rate parameter
TH = temperature coefficient for reaction rate correction (typically about 1.06)
TMP = soil layer temperature in degrees C

When temperatures are greater than 35 degrees C, the rate is considered optimum, that is, KK is set equal to K. When the temperature of the soil layer is below 4 degrees C or the layer is dry, no biochemical transformations occur.

Identifiers with a leading "K" (eg. KDNI) are the optimum rates; those for corrected rates have both a leading and trailing "K" (eg. KDNIK).

The corrected reaction rate parameters are determined every biochemical reaction interval and multiplied by the respective storages as shown in Figure 4.2(1).10-1 to obtain the reaction fluxes. Plant uptake can vary monthly and can be distributed between nitrate and ammonium by the parameters NO3UTF and NH4UTF. These parameters are intended to designate the fraction of plant uptake from each species of N; the sum of NO3UTF and NH4UTF should be 1.0.

The first order reaction rate fluxes that are shown in Figure 4.2(1).10-1 are coupled, that is, added to and subtracted from the storages simultaneously. The coupling of the fluxes is efficient in use of computer time but has a tendency to produce unrealistic negative storages when large reaction intervals and large reaction rates are used jointly. A method has been introduced which will modify the reaction fluxes so that they do not produce negative storages. A warning message is issued when this modification occurs.

4.2(1).11 Simulate Phosphorous Behavior in Detail (Section PHOS of Module PERLND)

Purpose

Module section PHOS simulates the behavior of phosphorus in a pervious land segment. This involves modeling the transport, plant uptake, adsorption/desorption, immobilization, and mineralization of the various forms of phosphorus. Because phosphorus is readily tied to soil and

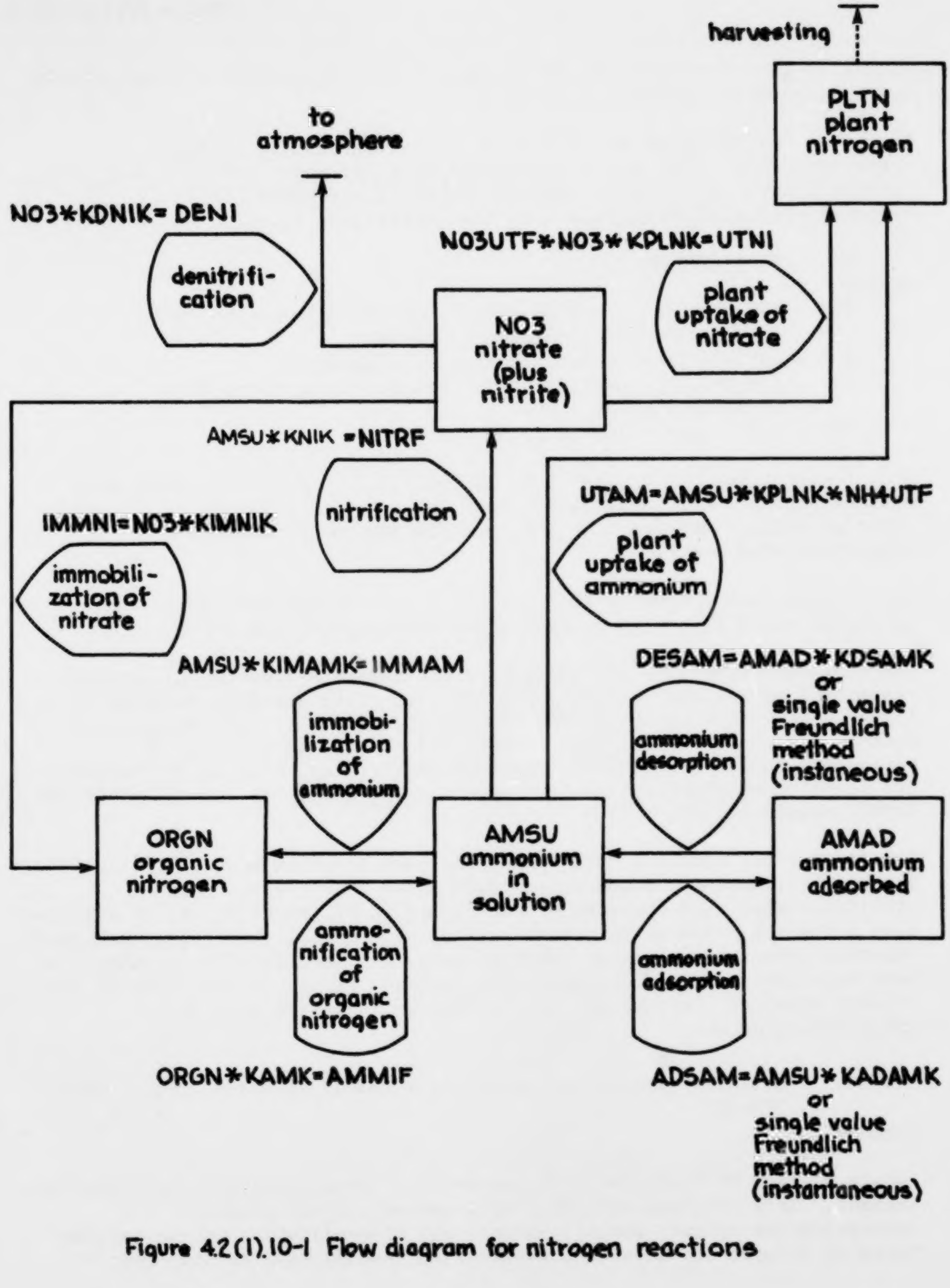


Figure 4.2(1).10-1 Flow diagram for nitrogen reactions

sediment, it is usually scarce in streams and lakes. In fact, in many cases it is the limiting nutrient in the eutrophication process. Because of its scarcity accurate simulation is particularly important.

Method of Simulating Phosphorus

The method used to transport and react phosphorus is the same as that used for nitrogen in module section NITR. The subroutines used to transport phosphorus are described in module section PEST. Organic phosphorus and adsorbed phosphate are removed on or with sediment by calling subroutine SEDMOV. Phosphate in solution is transported in the moving water using subroutines TOPMOV and SUBMOV. Phosphorus reaction is simulated in the soil by subroutine PHORXN.

In subroutine PHORXN, phosphate is adsorbed and desorbed by either first order kinetics or by the Freundlich method. The mechanics of these methods are described in module section PEST. As with the simulation of ammonium adsorption/desorption, the frequency of this chemical reaction for phosphate can also be specified. Unlike ammonium, typically phosphate includes a large portion which is not attached to the soil particle but is combined with cations. This is because phosphate is much less soluble with the ions found in soils than ammonium.

Other reactions performed by subroutine PHORXN include mineralization, immobilization, and plant uptake. These are accomplished using temperature dependent first order kinetics; the same method used for the nitrogen reactions. The general description of this process is in module section NITR. Figure 4.2(1).11-1 shows the parameters and equations used to calculate the reaction fluxes for phosphorus. Reactions are simulated for each of the four soil layers using separate parameter sets for each layer. As with nitrogen, the biochemical phosphate reaction fluxes of mineralization, immobilization, and plant uptake can be determined at an interval less frequent than the basic simulation interval.

4.2(1).12 Simulate Movement of a Tracer (Section TRACER of Module PERLND)

Purpose

The purpose of this code is to simulate the movement of any nonreactive tracer (conservative) in a pervious land segment. Chloride, bromide, and dyes are commonly used tracers which can be simulated by section TRACER. Also, total dissolved salts could possibly be modeled by this section. Typically, this code is applied to chloride to calibrate solute movement through the soil profile. This involves adjustment of the percolation retardation factors (see section MSTLAY) until good agreement with observed chloride concentrations has been obtained. Once these factors have been calibrated, they are used to simulate the transport of other solutes, such as nitrate.

Method of Simulating Tracer Transport

Tracer simulation uses the agri-chemical solute transport subroutines TOPMOV and SUBMOV which are described in section PEST. No reactions are modeled.

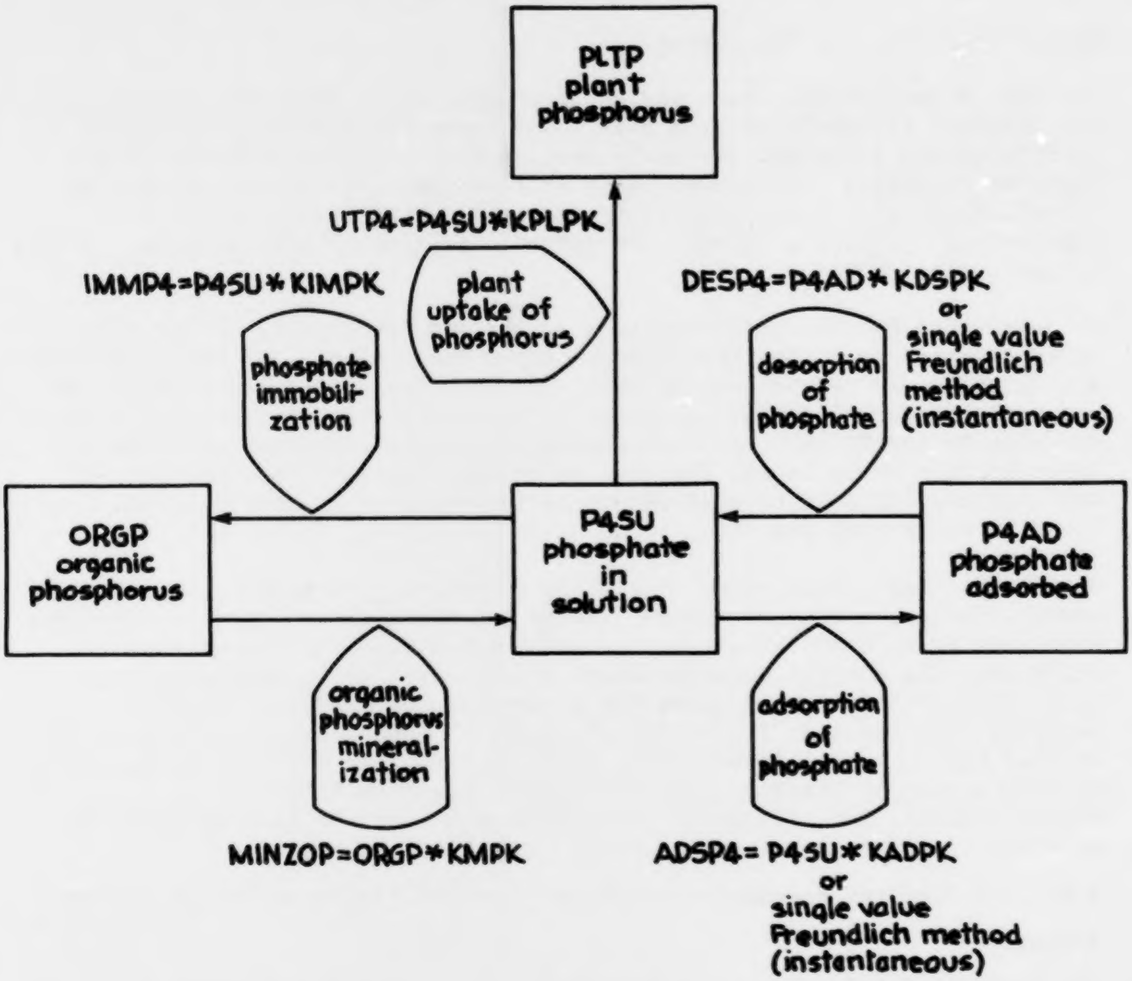


Figure 4.2(1).11-1 Flow diagram for phosphorus reactions

4.2(2) Simulate an Impervious Land Segment (Module IMPLND)

In an impervious land segment, little or no infiltration occurs. However, land surface processes do occur as illustrated in Figure 4.2(2)-1. Snow may accumulate and melt, and water may be stored or may evaporate. Various water quality constituents accumulate and are removed. Water, solids, and various pollutants flow from the segments by moving laterally to a downslope segment or to a reach/reservoir.

Module IMPLND simulates these processes. The sections of IMPLND and their functions are given in Structure Chart 4.2(2) (Part D). They are executed from left to right. Many of them are similar to the corresponding sections in the PERLND module. In fact, since sections SNOW and ATEMP perform functions that can be applied to pervious or impervious segments, they are shared by both modules. IWATER is analogous to PWATER in module PERLND; SOLIDS is analogous to SEDMNT; IWTGAS is analogous to PWTGAS; and IQUAL is analogous to PQUAL. However, the IMPLND sections are simpler since they contain no infiltration function and consequently no subsurface flows. IPTOT, IBAROT, and IPRINT service the IMPLND module similarly to the corresponding code in PERLND.

4.2(2).3 Simulate the Water Budget for an Impervious Land Segment (Section IWATER of Module IMPLND)

Purpose

Section IWATER simulates the retention, routing, and evaporation of water from an impervious land segment.

Method

Section IWATER is similar to section PWATER of the PERLND module. However, IWATER is simpler because there is no infiltration and consequently no subsurface processes. IWATER is composed of the parent subroutine plus three subordinate subroutines: RETN, IROUTE, and EVRETN. RETN is analogous to ICEPT, IROUTE is analogous to PROUTE, and EVRETN is analogous to EVICEP in module section PWATER. The time series requirements are the same as for section PWATER.

Figure 4.2(2).3-1 schematically represents the fluxes and storages simulated in module section IWATER. Moisture (SUPY) is supplied by precipitation, or under snow conditions, it is supplied by the rain not falling on the snowpack plus the water yielded by the snowpack. This moisture is available for retention; subroutine RETN performs the retention functions. Lateral surface inflow (SURLI) may also be retained if the user so specifies by setting the flag parameter RTLIFG=1. Otherwise, retention inflow (RETI)

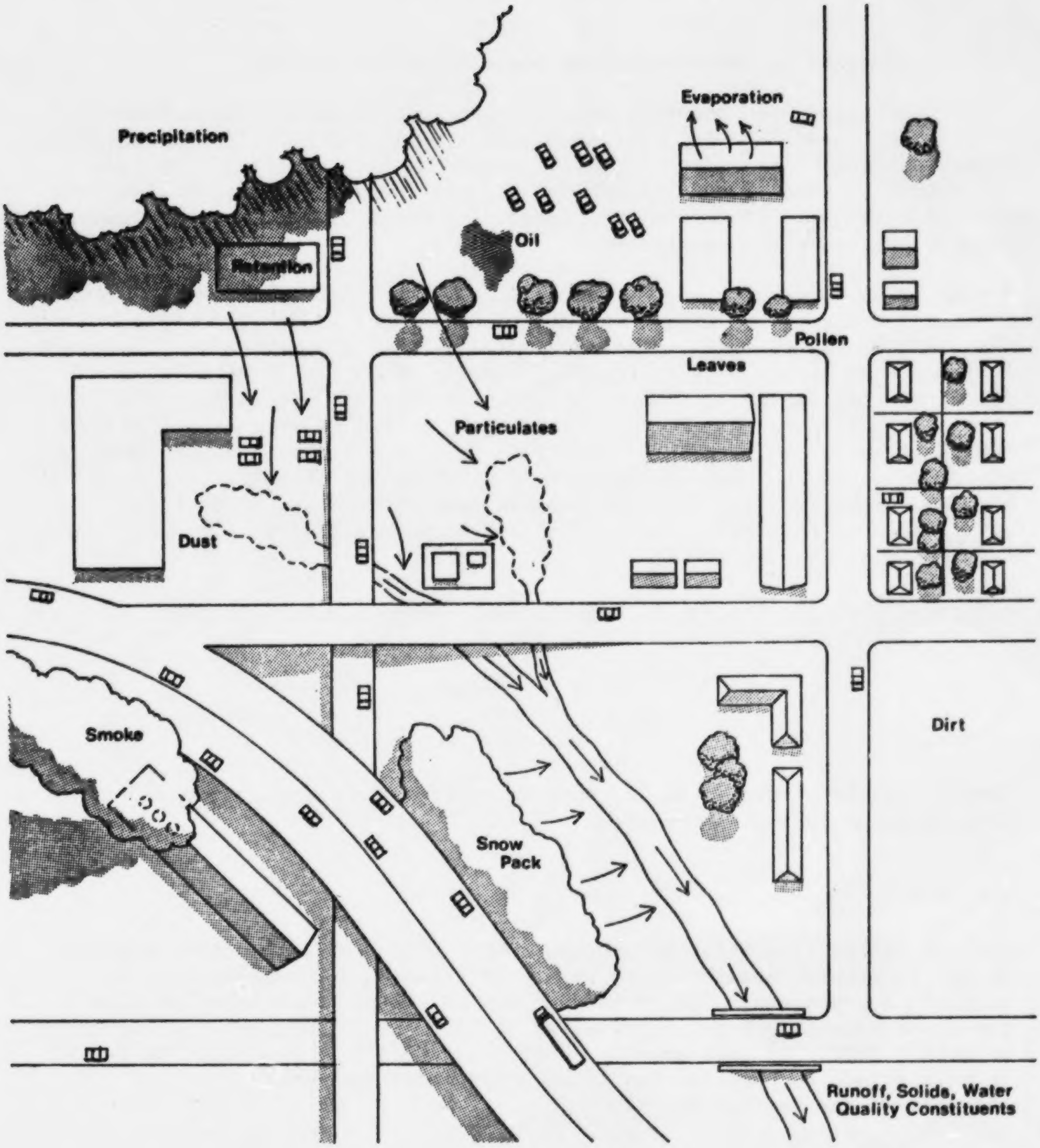


Figure 4.2(2)-1 Impervious land segment processes

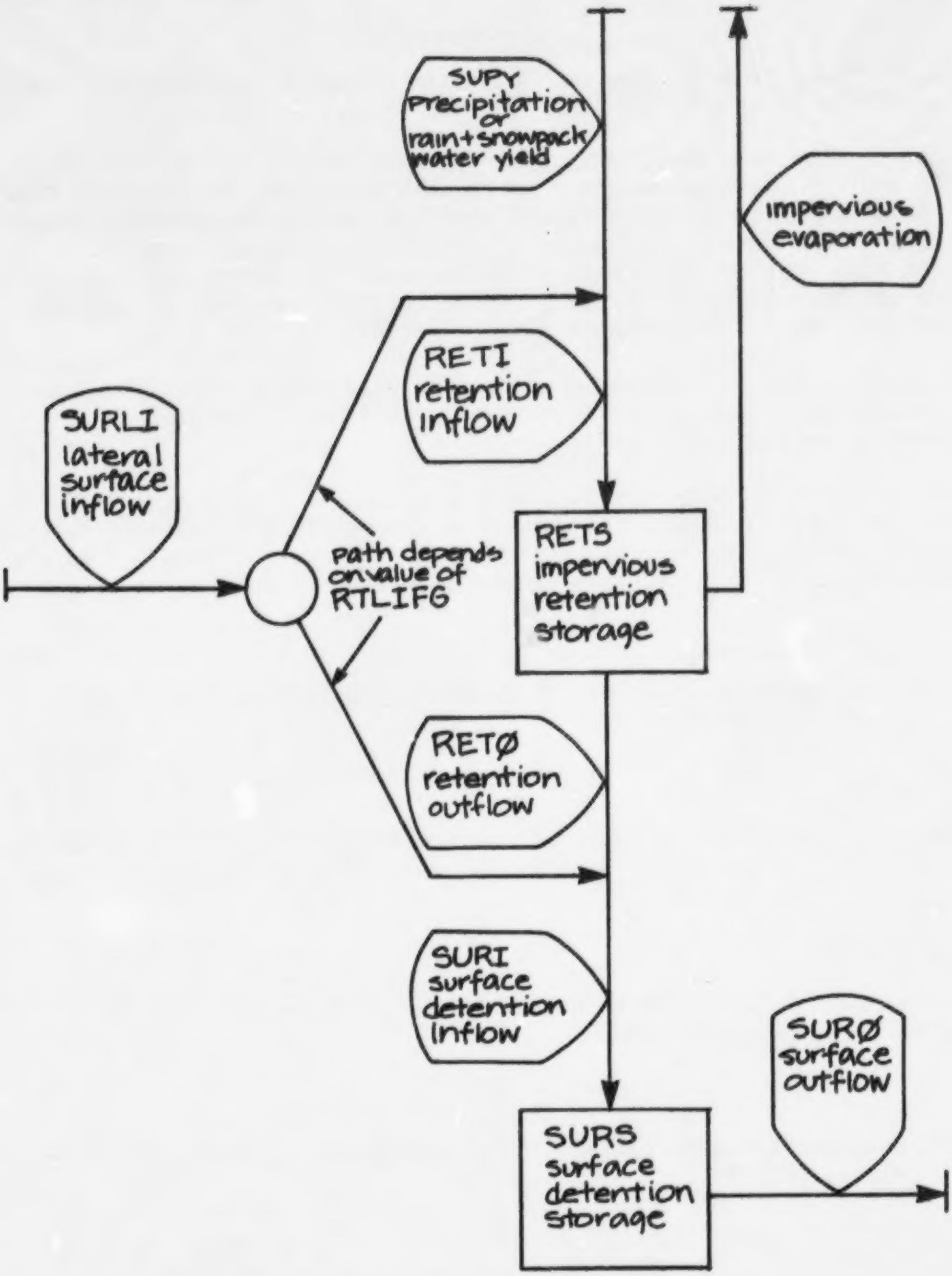


Figure 4.2(2).3-1 Hydrological processes

equals SUPY. Moisture exceeding the retention capacity overflows the storage and is available for runoff.

The retention capacity, defined by the parameter RETSC, can be used to designate any retention of moisture which does not reach the overland flow plane. RETSC may be used to represent roof top catchments, asphalt wetting, urban vegetation, improper drainage, or any other containment of water that will never flow from the land segment. The user may supply the retention capacity on a monthly basis to account for seasonal variations, or may supply one value designating a fixed capacity.

Water held in retention storage is removed by evaporation (IMPEV). The amount evaporated is determined in subroutine EVRETN. Potential evaporation is an input time series.

Retention outflow (RETO) is combined with any lateral inflow when RTLIFG=0 producing the total inflow to the detention storage (SURI). Water remaining in the detention storage plus any inflow is considered the moisture supply. The moisture supply is routed from the land surface in subroutine IROUTE.

4.2(2).3.2 Determine How Much of the Moisture Supply Runs Off (subroutine IROUTE)

Purpose

The purpose of subroutine IROUTE is to determine how much of the moisture supply runs off the impervious surface in one simulation interval.

Method of Routing

A method similar to that used in module PERLND (Section 4.2(1).3.2.1.3) is employed to route overland flow.

4.2(2).4 Simulate Accumulation and Removal of Solids (Section SOLIDS of Module IMPLND)

Purpose

Module section SOLIDS simulates the accumulation and removal of solids by runoff and other means from the impervious land segment. The solids outflow may be used in section IQUAL to simulate quality constituents associated with particulates.

Method

The equations used in this section are based on those in the NPS Model (Donigian and Crawford 1976b). Figure 4.2(2).4-1 schmatically represents the fluxes and storages simulated by section SOLIDS. Lateral input of solids by water flow is a user designated option which is unique to HSPF. Washoff of solids may be simulated by one of two ways. One subroutine is similar to the method used in the NPS Model. However, this method is dimensionally nonhomogeneous. That is, a flux and a storage are added making the answer more interval dependent. This technique has only been used with 15-min intervals. The other subroutine is dimensionally homogenous, since only a flux term is used in the solution. However, this method has not been tested:

The accumulation and removal of solids which occurs independently of runoff (eg. by atmospheric fallout, street cleaning) is handled in subroutine ACCUM.

4.2(2).4.1 Washoff Solids Using Method 1 (subroutine SOSLD1)

Purpose

Subroutines SOSLD1 and SOSLD2 perform the same task but by different methods. They simulate the washoff of solids from an impervious land segment.

Method

When simulating the washoff of solids, the transport capacity of the overland flow is estimated and compared to the amount of solids available. The transport capacity is calculated by the equation:

$$STCAP = DELT60*KEIM*((SURS + SURO)/DELT60)**JEIM \qquad (1)$$

where:

STCAP = capacity for removing solids in tons/acre per interval
DELT60 = hours per interval
KEIM = coefficient for transport of solids
SURS = surface water storage in inches
SURO = surface outflow of water in in./interval
JEIM = exponent for transport of solids

When STCAP is greater than the amount of solids in storage, washoff is calculated by:

$$SOSLD = SLDS*SURO/(SURS + SURO) \qquad (2)$$

If the storage is sufficient to fulfill the transport capacity, then the following relationship is used:

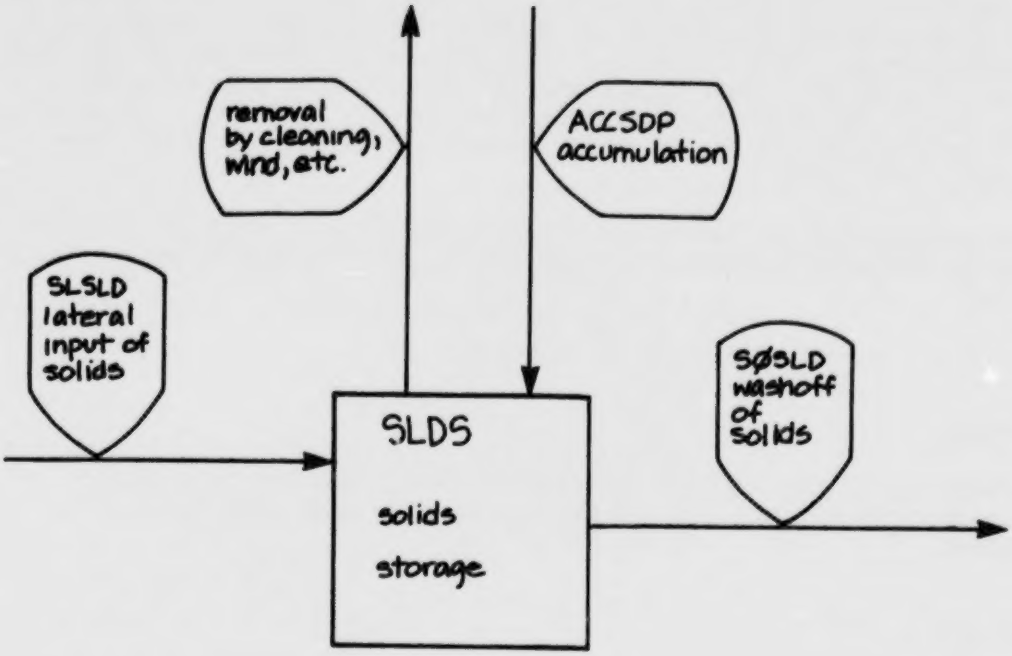


Figure 4.2(2).4-1 Flow diagram of the SOLIDS section of the IMPLND application module.

SOSLD = STCAP*SURO/(SURS + SURO) (3)

where:

SOSLD = washoff of solids in tons/acre per interval
SLDS = solids storage in tons/acre

SOSLD is then subtracted from SLDS.

Subroutine SOSLD1 differs from SOSLD2 in that it uses the dimensionally nonhomogeneous term (SURS + SURO)/DELT60 in the above equations, while SOSLD2 uses the homogeneous term SURO/DELT60.

4.2(2).4.2 Washoff Solids Using Method 2 (subroutine SOSLD2)

Purpose

The purpose of this subroutine is the same as SOSLD1. They only differ in method.

Method of Determining Removal

This method of determining sediment removal has not been tested. Unlike subroutine SOSLD1, it makes use of the dimensionally homogeneous term SURO/DELT60 instead of (SURO+SURS)/DELT60 in the following equation.

STCAP = DELT60*KEIM*(SURO/DELT60)**JEIM (4)

When STCAP is more than the amount of solids in storage, the flow washes off all of the solids storage (SLDS). However, when STCAP is less than the amount of solids in storage, the situation is transport limiting, so SOSLD is equal to STCAP.

4.2(2).4.3 Accumulate and Remove Solids Independently of Runoff (subroutine ACCUM)

Purpose

Subroutine ACCUM simulates the accumulation and removal of solids independently of runoff; for example, atmospheric fallout and street cleaning.

Method

The storage is updated once a day, on those days when precipitation did not occur during the previous day, using the equation:

SLDS = ACCSDP + SLDSS*(1.0 - REMSDP) (5)

where:

ACCSDP = accumulation rate of the solids storage, lbs/acre per day
SLDS = solids in storage at end of interval in tons/acre
SLDSS = solids in storage at start of interval in tons/acre
REMSDP = unit removal rate of solids in storage (i.e. fraction removed per day)

ACCSDP and REMSDP may be input on a monthly basis to account for seasonal variations.

Note that, if no runoff occurs, equation 5 will cause the solids storage to asymptotically approach a limiting value. The limit, found by setting SLDS and SLDSS to the same value (SLDSL), is:

SLDSL = ACCSDP/REMSDP (6)

4.2(2).5 Estimate Water Temperature and Dissolved Gas Concentrations (Section IWTGAS of Module IMPLND)

Purpose

IWTGAS estimates the water temperature and concentrations of dissolved oxygen and carbon dioxide in the outflow from the impervious land segment.

Method

Outflow temperature is estimated by the following regression equation:

SOTMP = AWTF + BWTF*AIRTC (1)

where:

SOTMP = impervious surface runoff temperature in degrees C
AWTF = Y-intercept
BWTF = slope
AIRTC = air temperature in degrees C

The parameters AWTF and BWTF may be input on a monthly basis. When snowmelt contributes to the outflow, SOTMP is set equal to 0.5.

The dissolved oxygen and carbon dioxide concentrations of the overland flow are assumed to be at saturation and are calculated as direct functions of water temperature. IWTGAS uses the following empirical nonlinear equation to relate dissolved oxygen at saturation to water temperature (Committee on Sanitary Engineering Research 1960):

SODOX = (14.652 + SOTMP*(-0.41022 +
SOTMP*(0.007991 - 0.000077774*SOTMP)))*ELEVGC (2)

where:

SODOX = concentration of dissolved oxygen in surface outflow in mg/l
SOTMP = surface outflow temperature in degrees C
ELEVGC = correction factor for elevation above sea level (ELEVGC is calculated by the Run Interpreter dependent upon mean elevation of the segment)

The empirical equation for dissolved carbon dioxide concentration of the overland flow (Harnard and Davis 1943) is:

```
SOCO2 = (10**(2385.73/ABSTMP - 14.0184 + 0.0152642*ABSTMP))
                  *0.000316*ELEVGC*12000.0                (3)
```

where:

SOCO2 = concentration of dissolved carbon dioxide in surface outflow in mg C/l
ABSTMP = absolute temperature of surface outflow in degrees K

4.2(2).6 Simulate Washoff of Quality Constituents Using Simple Relationships with Solids and Water Yield (Section IQUAL of Module IMPLND)

Purpose

The IQUAL module section simulates water quality constituents or pollutants in the outflows from an impervious land segment using simple relationships with water yield and/or solids. Any constituent can be simulated by this module section. The user supplies the name, units and parameter values appropriate to each of the constituents that he wishes to simulate. Note that more detailed methods of simulating solids, heat, dissolved oxygen, and dissolved carbon dioxide removal from the impervious land segment are available in other sections of IMPLND.

Approach

The basic algorithms used to simulate quality constituents are a synthesis of those used in the NPS Model (Donigian and Crawford 1976b) and HSP QUALITY (Hydrocomp 1977). However, some options and combinations are unique to HSPF.

Figure 4.2(2).6-1 shows schematically the fluxes and storages represented in module section IQUAL. A quality constituent may be simulated by two methods. One approach is to simulate the constituent by association with solids removal. The other approach is to simulate it by using basic accumulation and depletion rates together with depletion by washoff; that is, constituent outflow from the surface is a function of the water flow and the constituent in storage. A combination of the two methods may be used in which the individual outfluxes are added to obtain the total surface outflow. These approaches will be discussed further in the descriptions of the corresponding subroutines.

IQUAL allows the user to simulate up to 10 quality constituents at a time. If a constituent is considered to be associated with solids, it is called a QUALSD. The corresponding term for constituents associated directly with overland flow is QUALOF. Each of the 10 constituents may be defined as either a QUALSD or a QUALOF or both. However, no more than seven of any one of the constituent types (QUALSD or QUALOF) may be simulated in one operation. The program uses a set of flag pointers to keep track of these associations. For example, QSDFP(3)=0 means that the third constituent is not associated with solids, whereas QSDFP(6)=4 means that the sixth constituent is the fourth solids associated constituent (QUALSD). Similar flag pointer arrays are used to indicate whether or not a quality constituent is a QUALOF.

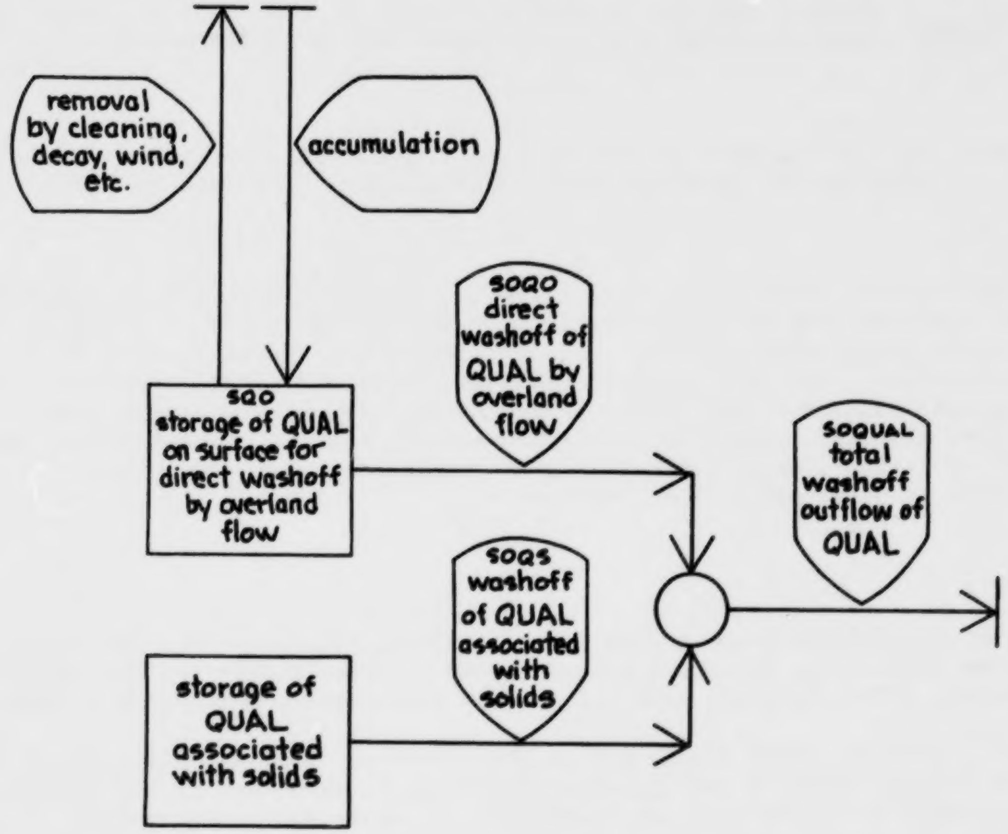

removal by cleaning, decay, wind, etc.
accumulation
SQO storage of QUAL on surface for direct washoff by overland flow
SOQO direct washoff of QUAL by overland flow
SOQUAL total washoff outflow of QUAL
SOQS washoff of QUAL associated with solids
storage of QUAL associated with solids

Figure 4.2(2).6-1 Flow diagram for IQUAL section of IMPLND Application Module

4.2(2).6.1 Remove by Association with Solids (subroutine WASHSD)

Purpose

WASHSD simulates the removal of a quality constituent from the impervious land surface by association with the solids removal determined in module section SOLIDS.

Method

This approach assumes that the particular quality constituent removed from the land surface is in proportion to the solids removal. The relation is specified by user-input "potency factors." Potency factors indicate the constituent strength relative to the solids removal from the surface. For each quality constituent associated with solids, the user supplies separate potency factors. The user is also able to supply monthly potency factors for constituents that vary somewhat consistently throughout the year.

Removal of the solids associated constituent by solids washoff is simulated by:

$$SOQS = SOSLD*POTFW \qquad (1)$$

where:

SOQS = flux of quality constituent associated with solids washoff in quantity/acre per interval
SOSLD = washoff of detached solids in tons/acre per interval
POTFW = washoff potency factor in quantity/ton

The unit "quantity" refers to mass units (pounds or tons in the English system) or some other quantity, such as number of organisms for coliforms. The user specifies the units of "quantity".

4.2(2).6.2 Accumulate and Remove by a Constant Unit Rate and by Overland Flow (subroutine WASHOF)

Purpose

WASHOF simulates the accumulation of a quality constituent on the impervious land surface and its removal by a constant unit rate and by overland flow.

Method

This subroutine differs from subroutine WASHSD in that the storage of the quality constituent is simulated. The stored constituent can be accumulated and removed by processes which are independent of storm events, such as cleaning, decay, and wind deposition, and it is washed off by overland flow.

The accumulation and removal rates can have monthly values to account for seasonal fluctuations.

When there is surface outflow and some quality constituent is in storage then washoff is simulated using the commonly used relationship:

$$\mathrm{SOQO} = \mathrm{SQO}*(1.0 - \mathrm{EXP}(-\mathrm{SURO}*\mathrm{WSFAC})) \tag{2}$$

where:

SOQO = washoff of the quality constituent from the land surface in quantity/acre per interval
SQO = storage of the quality constituent on the surface in quantity/acre
SURO = surface outflow of water in in./interval
WSFAC = susceptibility of the quality constituent to washoff in units of 1/inch
EXP = Fortran exponential function

The storage is updated once a day to account for accumulation and removal which occurs independent of runoff by the equation:

$$\mathrm{SQO} = \mathrm{ACQOP} + \mathrm{SQOS}*(1.0 - \mathrm{REMQOP}) \tag{3}$$

where:

ACQOP = accumulation rate of the constituent, quantity/acre per day
SQOS = SQO at the start of the interval
REMQOP = unit removal rate of the stored constituent, per day

The Run Interpreter computes REMQOP and WSFAC for this subroutine according to:

$$\mathrm{REMQOP} = \mathrm{ACQOP}/\mathrm{SQOLIM} \tag{4}$$

where:

SQOLIM = asymptotic limit for SQO as time approaches infinity, (quantity/acre), if no washoff occurs

and

$$\mathrm{WSFAC} = 2.30/\mathrm{WSQOP} \tag{5}$$

where:

WSQOP = rate of surface runoff which results in a 90 percent washoff in one hour, in./hr

Since the unit removal rate (REMQOP) is computed from two other parameters, it is not supplied directly by the user.

4.2(3) Simulate a Free-flowing Reach or Mixed Reservoir (Module RCHRES)

This module simulates the processes which occur in a single reach of open or closed channel or a completely mixed lake. For convenience such a processing unit is referred to as a RCHRES throughout this documentation. In keeping with the assumption of complete mixing, the RCHRES consists of a single zone situated between two nodes, which are the extremities of the RCHRES.

Flow through a RCHRES is assumed to be unidirectional. The inflow and outflow of materials through a RCHRES are illustrated in Figure 4.2(3)-1. Water and other constituents which arrive from other RCHRES's and local sources enter the RCHRES through a single gate (INFLO). Outflows may leave the RCHRES through one of several gates (OFLO). A RCHRES can have up to 5 OFLO gates. Precipitation, evaporation, and other fluxes also influence the processes which occur in the RCHRES but do not pass through the gates.

The ten major subdivisions of the RCHRES module and their functions are presented in Structure Chart 4.2(3). RPTOT, RBAROT, and RPRINT perform the storage and printout of results from the other module sections of RCHRES (HYDR through RQUAL). Within a module section, simulation of physical processes (longitudinal advection, sinking, benthal release) is always performed before simulation of biochemical processes.

The user specifies which module sections are active. If any "quality" sections (CONS through RQUAL) are active, section ADCALC must also be active; it computes certain quantities needed to simulate advection of the quality constituents. Besides fulfilling this requirement, the user must ensure that all the time series required by the active sections are available, either as supplied input time series or as data computed by another module section. For example, if RQUAL is active, the water temperature must be supplied, either as an input time series or by activating section HTRCH which will compute it.

4.2(3).1 Simulate Hydraulic Behavior (Section HYDR of Module RCHRES)

Purpose

The purpose of this code is to simulate the hydraulic processes occurring in a reach or a mixed reservoir (RCHRES). The final goal of the process may be to route floods, study reservoir behavior, or analyze constituents dissolved in the water.

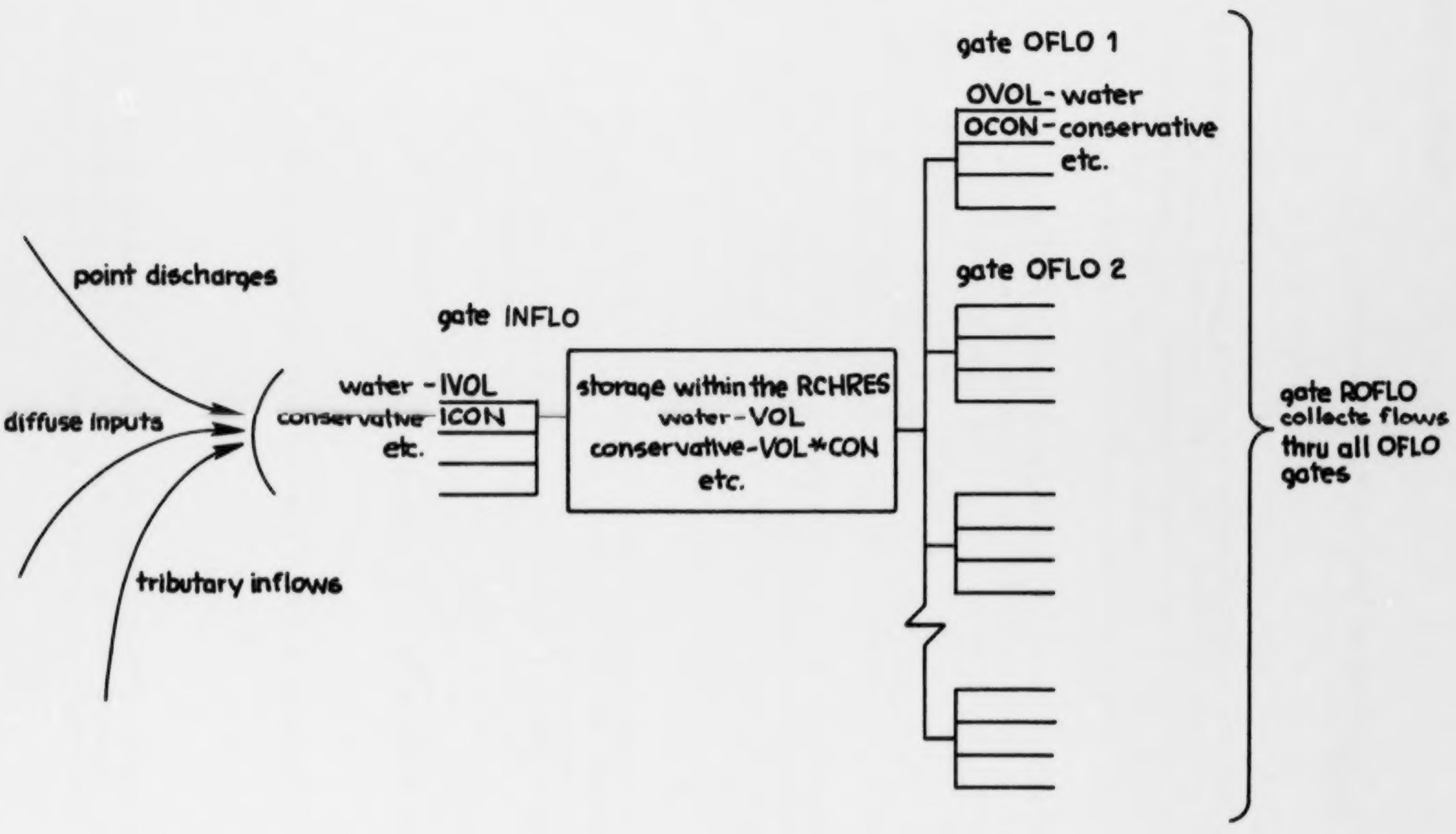


Figure 4.2(3)-1 Flow of materials through a RCHRES

Schematic View of Fluxes and Storage

Figure 4.2(3).1-1 shows the principal state variable (stored volume) and fluxes with which this part of HSPF deals.

All water entering the RCHRES from surface and subsurface sources arrives through "gate" INFLO; this quantity is called IVOL. The user indicates the time series which enter this gate in the EXT SOURCES or NETWORK Blocks of his User's Control Input (UCI). If no time series are specified, the system assumes the RCHRES has zero inflow.

The volume of water which leaves the RCHRES during a simulation time interval, through gate OFLO(N), is called OVOL(N). The total outflow is ROVOL.

The input of water from precipitation falling directly on the water surface and the loss of water by evaporation from the surface can also be considered. The user activates these options by supplying the time series PREC and/or POTEV in his User's Control Input (external sources block). These time series are in units of depth/interval. The code multiplies these quantities by the current surface area of the RCHRES to obtain volumes of input/output. If either time series is absent from the UCI it is assumed that the option is inactive and the corresponding flux is zero.

The basic equation is that of continuity:

$$\mathrm{VOL} - \mathrm{VOLS} = \mathrm{IVOL} + \mathrm{PRSUPY} - \mathrm{VOLEV} - \mathrm{ROVOL} \qquad (1)$$

where:
VOL = volume at the end of the interval
VOLS = volume at the start of the interval

This can be written as:

$$\mathrm{VOL} = \mathrm{VOLT} - \mathrm{ROVOL} \qquad (2)$$

where:
VOLT = IVOL + PRSUPY - VOLEV + VOLS

The principal task of this subroutine is to estimate ROVOL and, hence, the volume at the end of the interval (VOL).

Calculation of Outflows and VOL

If water is available, it is assumed that the total volume of water leaving a RCHRES in an interval is:

$$\mathrm{ROVOL} = (\mathrm{KS}*\mathrm{ROS} + \mathrm{COKS}*\mathrm{ROD})*\mathrm{DELTS} \qquad (3)$$

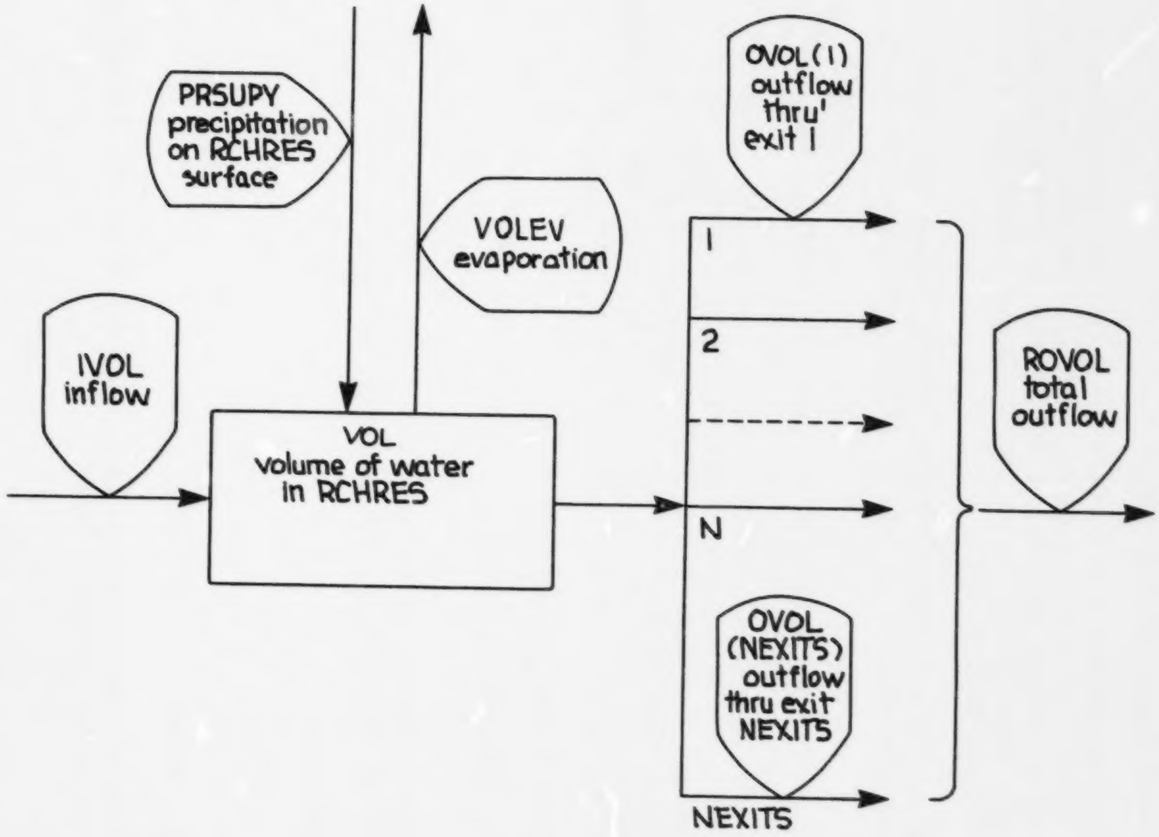

PRSUPY precipitation on RCHRES surface
VOLEV evaporation
IVOL inflow
VOL volume of water in RCHRES
OVOL(1) outflow thru' exit 1
1
2
N
NEXITS
OVOL (NEXITS) outflow thru exit NEXITS
ROVOL total outflow

Figure 4.2(3).1-1 Flow diagram for the HYDR Section of the RCHRES Application Module

where:
KS = weighting factor $(0 \le KS \le 0.99)$
COKS = 1.0 - KS (complement of KS)
ROS = total rate of outflow from the RCHRES at the start of the interval
ROD = total rate of demanded outflow for the end of the interval
DELTS = simulation interval in seconds

That is, the mean rate of outflow is assumed to be a weighted mean of the rates at the start and end of the interval. The weighting factor KS is supplied either by the user or by default. Care should be exercised in selecting a value because, as KS increases from 0.0 to 1.0, there is an increasing risk that the computation of outflow rates will become unstable. Theoretically, a value of 0.5 gives the most accurate results, provided oscillations do not occur. The default value of 0.0 has zero risk, but gives less accurate results. Users are advised to be very careful if a nonzero value is used; it seems that one is never justified in selecting a value greater than 0.5.

Combination of Equations 2 and 3 yields:

$$\mathrm{VOL} = \mathrm{VOLT} - (\mathrm{KS}*\mathrm{ROS} + \mathrm{COKS}*\mathrm{ROD})*\mathrm{DELTS} \qquad (4)$$

There are two unknown values in this equation: VOL and ROD. Thus, a second relation is required to solve the problem. To provide this function, it is assumed that outflow demands for the individual exits are of the form:

$$\begin{aligned}
&\mathrm{OD}(1) = \mathrm{f1}(\mathrm{VOL},t)\\
&\mathrm{OD}(2) = \mathrm{f2}(\mathrm{VOL},t)\\
&\cdot\\
&\cdot\\
&\mathrm{OD}(\mathrm{NEXITS}) = \mathrm{fNEXITS}(\mathrm{VOL},t) \qquad (5)
\end{aligned}$$

That is, the outflow demand for each exit is a function of volume or of time or a combination. This topic is discussed in greater detail in Section 4.2(3).1.1.2.

It follows that the total outflow demand is of similar form:

$$\mathrm{ROD} = \mathrm{funct}(\mathrm{VOL},t) \qquad (6)$$

At a given time in the simulation, t is known and the above functions reduce to:

$$\begin{aligned}
&\mathrm{OD}(N) = \mathrm{fN}(\mathrm{VOL}) &(7)\\
&\mathrm{ROD} = \mathrm{funct}(\mathrm{VOL}) &(8)
\end{aligned}$$

Equation 8 provides the second relation required to solve the problem. Equations 4, 7, and 8 are shown in Figure 4.2(3).1-2. The point of intersection of Equations 4 and 8 gives the values RO, VOL, and hence O(1), O(2), etc.

where:

RO = total rate of outflow from the RCHRES at the end of the interval

O(N) = rate of outflow through exit N at the end of the interval

In HSPF, it is assumed that each outflow demand can be represented by one or both of the following types of components:

> Component = function(VOL). This is most useful in simulating RCHRES's where there is no control over the flow or where gate settings are only a function of water level.
>
> Component = function(time). This is most useful for handling demands for municipal, industrial, or agricultural use. The function may be cyclic (for example, annual cycle) or general (for example, annual cycle superimposed on an increasing trend). The user must supply the component in the form of an input time series.

If a user indicates that both types of component are present in an outflow demand, then he must also specify how they are to be combined to get the demand. HSPF allows the following options:

1. OD(N) = Min [fN(VOL),gN(t)]. This is useful in cases such as the following:

   Suppose a water user has an optimum demand which may be expressed as a function of time (g(t)); however, his pump has a limited capacity to deliver water. This capacity is a function of the water level in the RCHRES from which the pump is drawing the water. Thus, it can be expressed as a function of the volume in the RCHRES (f(VOL)). Then, his actual demand for water will be the minimum of the two functions.

2. OD(N) = Max [fN(VOL),gN(t)]

3. OD(N) = fN(VOL) + gN(t)

If one or more outflow demands have an f(VOL) component (Fig.4.2(3).1-2a), subroutine ROUTE is called to solve the routing equations. In this case, the evaluation of the outflow demands and the solution of the equations can be quite complicated.

If there is no f(VOL) component in any demand, the process is much simpler (Figure 4.2(3).1-2b). Subroutine NOROUT is called in this case.

Representing the Geometry and Hydraulic Properties of a RCHRES

HSPF makes no assumptions regarding the shape of a RCHRES. It does not require that the cross section be trapezoidal or even that the shape be prismoidal. This is one reason why both free flowing reaches and reservoirs can be handled by the same application module. Both of the shapes shown in Figure 4.2(3).1-3a are acceptable. However, HSPF does assume that:

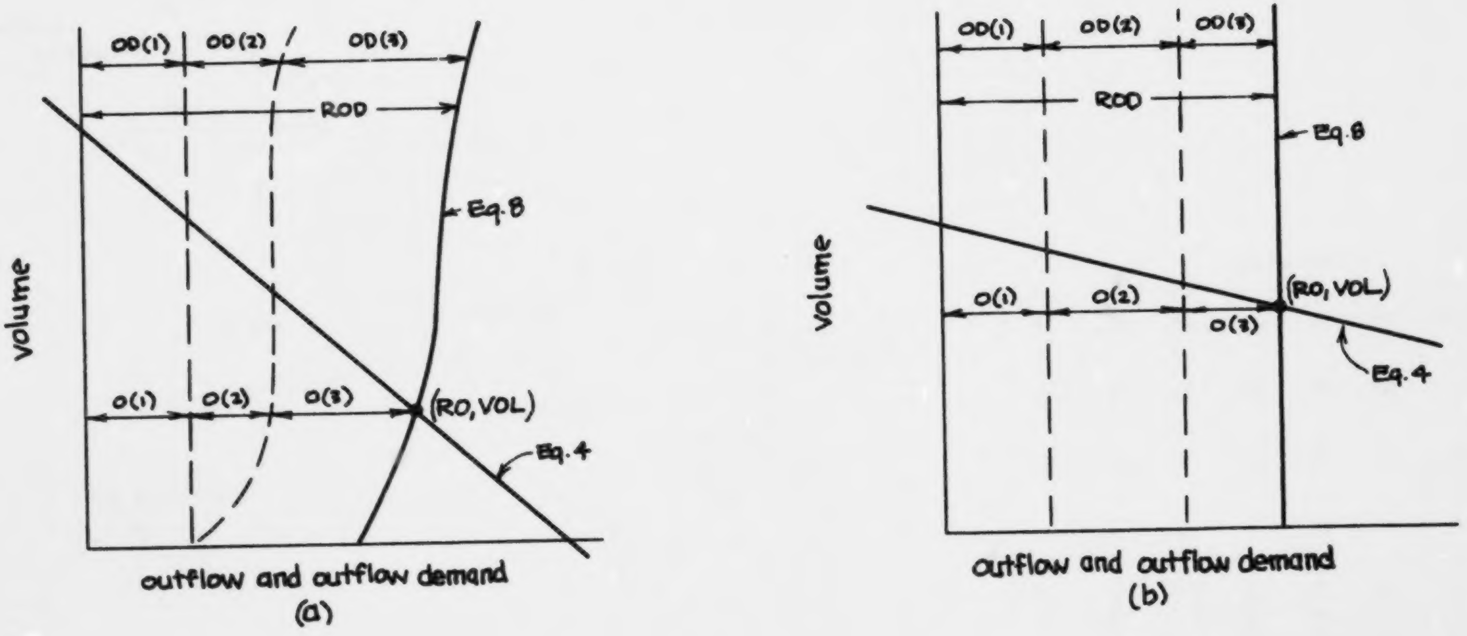
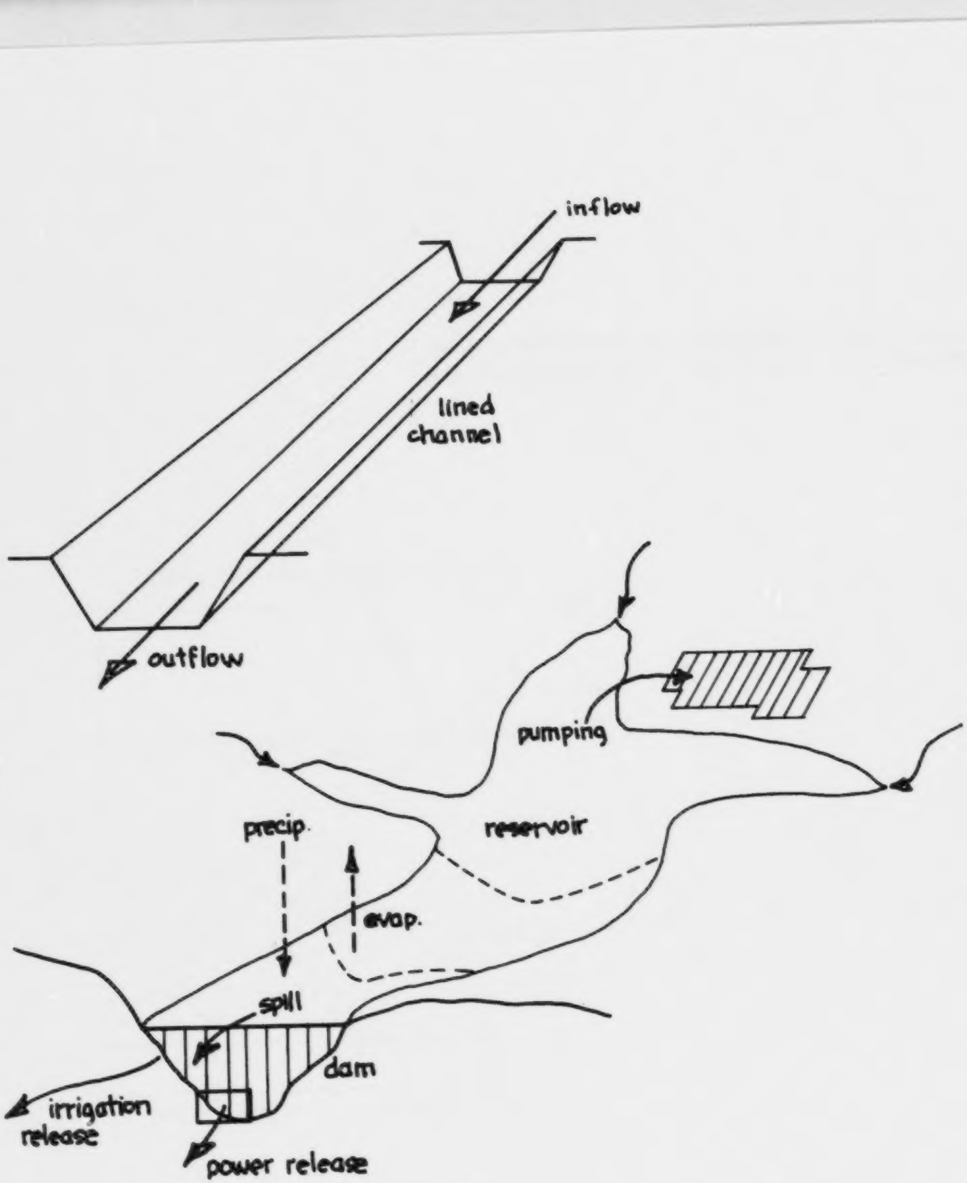


Figure 4.2(3).1-2 Graphical representation of the equations used to compute outflow rates and volume.



(a) Typical reach and mixed reservoir

col	1	2	3	4	5	6	7
row #	depth	surface area	volume	f1 (vol)	f2 (vol)	f3 (vol)	f4 (vol)
1	0	0	0	10	0	5	0
2	1.5	1	.8	12	6	10	0
3	10.0	15	80	12	18	10	0
4	50.0	100	2500	12	36	20	20

RCHTAB

(b) Function table used to specify geometry and hydraulic properties of a RCHRES

Figure 4.2(3).1-3 Typical RCHRES configurations, and the method used to represent geometric and hydraulic properties

1. There is a fixed relation between depth (at the deepest point in the RCHRES), surface area, and volume.

2. For any outflow demand with an f(VOL) component, the functional relation is constant in time (with the exception discussed in Section 4.2(3).2.1.1).

These assumptions rule out cases where the flow reverses direction or where one RCHRES influences another upstream of it in a time-dependent way. No account is taken of momentum. The routing technique falls in the class known as "storage routing" or "kinematic wave" methods.

The user specifies the properties of a RCHRES in a table called RCHTAB (Figure 4.2(3).1-3b). It has columns for the depth, surface area, volume, and volume dependent functions (fN(VOL)). Each row contains values appropriate to a specified water surface elevation. The system obtains intermediate values by interpolation. Thus, the number of rows in RCHTAB depends on the size of the cross section and the desired resolution. The table is included in the User's Control Input, in the function tables (FTABLES) block. A subsidiary, stand alone program can be used to generate this table for RCHRES's with simple properties (for prismoidal channels with uniform flow, use Manning's equation).

Auxiliary Variables

Besides calculating outflow rates and the volume in a RCHRES, HSPF can compute the values of some auxiliary state variables:

1. If AUX1FG=1, DEP, STAGE, SAREA, AVDEP, and TWID are computed where: DEP is the depth at the deepest point; STAGE is the water stage at a related point; SAREA is the surface area of water in the RCHRES; AVDEP is the average depth (volume/surface area); TWID is the top width (surface area/length).

   This is done by subroutine AUXIL.

2. If AUX2FG=1, AVSECT and AVVEL are computed where: AVSECT is the average cross section (volume/length); AVVEL is the average velocity (discharge/AVSECT).

Note that these are point-valued time series; that is, they apply at the boundaries (start or end) of simulation time intervals.

The user specifies whether AUX1FG and AUX2FG are ON or OFF. If he is simulating certain constituents, one or both of these flags might be required to be ON. For example, simulation of oxygen (subroutine group OXRX) requires both flags ON.

4.2(3).1.1 Calculate Outflows Using Hydraulic Routing (subroutine ROUTE)

Purpose

ROUTE computes the rates and volumes of outflow from a RCHRES and the new volume in cases where at least one outflow demand has an f(VOL) component.

Method

The problem is to solve simultaneously Equations 4 and 8. The cases which arise are shown graphically on Figure 4.2(3).1-4. Equations 7 and 8 are represented by a series of straight line segments. The breakpoints in the lines correspond to a row of entries in RCHTAB. A segment of Equation 8 can be represented by the equation:

$$(VOL - V1)/(ROD - ROD1) = (V2 - V1)/(ROD2 - ROD1) \tag{9}$$

where V1,V2 are volumes specified in adjacent rows of RCHTAB, for the lower and upper extremities of the straight-line segment, respectively. ROD1,ROD2 are the corresponding total outflow demands.

The first step is to find the intercept of Equation 4 on the volume axis:

$$VOLINT = VOLT - KS*ROS*DELTS \tag{10}$$

If VOLINT is less than zero, the equations cannot be solved (case 3). Equation 4 will give a negative value for VOL, even if ROD is zero. Physically, this means that we started the interval with too little water to satisfy the projected outflow demand, even if the outflow rate at the end of the interval is zero. Accordingly, the code sets:

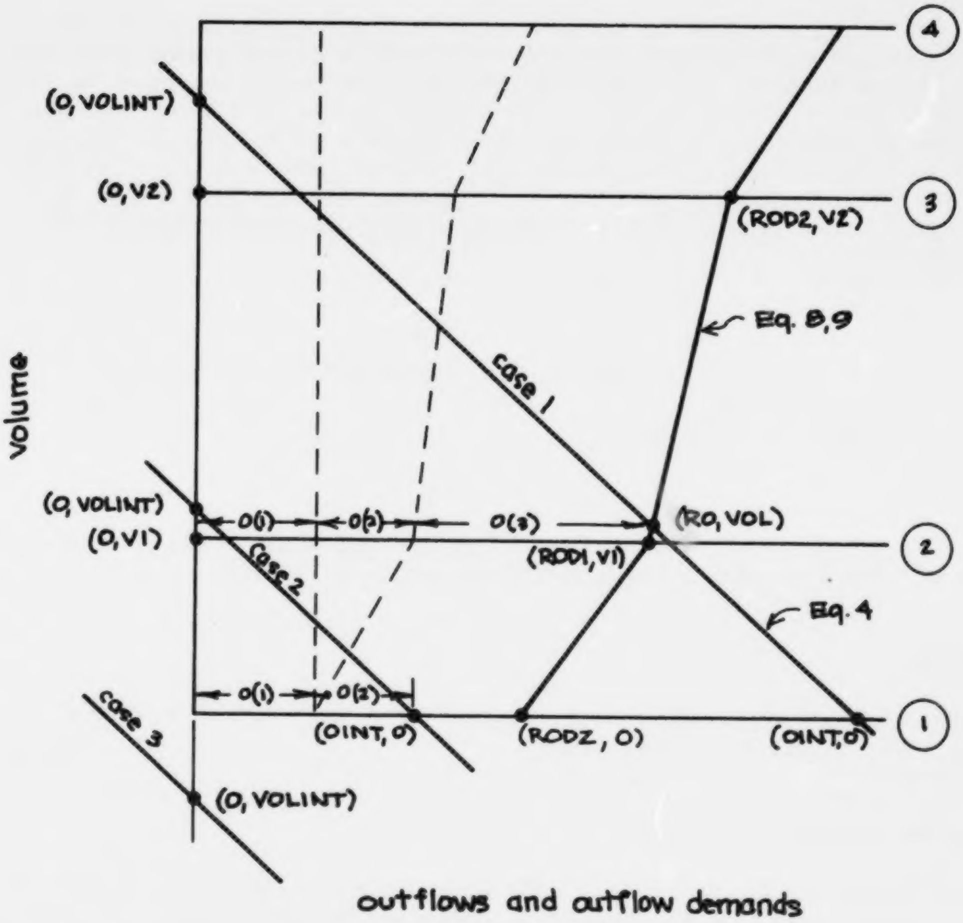
VOL = 0.0
RO = 0.0
O(*) = 0.0
ROVOL = VOLT

If VOLINT is greater than or equal to zero, the outflow rate at the end of the interval will be nonzero (case 1 or 2). To determine the case:

1. The intercept of Equation 4 on the Volume axis is found:

   $$OINT = VOLINT/(DELTS*COKS) \tag{11}$$

2. The maximum outflow demand for which the volume is still zero (RODZ) is found.



(–,–) are coordinates of points

② is row no. in RCHTAB which contains data for this level

Figure 4.2(5).1-4. Graphical representation of the work performed by subroutine ROUTE

If OINT is greater than RODZ, Equations 4 and 8 can be solved (case 1). The solution involves searching for the segment of Equation 8 which contains the point of intersection of the graphs, and finding the coordinates of the point (RO,VOL). This is done by subroutine SOLVE.

If OINT is less than or equal to RODZ, Equations 4 and 8 cannot be solved (case 2). Physically this means that the RCHRES will instantaneously go dry at the end of the interval with total outflow rate at that time equal to OINT. Accordingly, the code assigns a zero value to the RCHRES volume, and total outflow is equal to the intercept of Equation 4 on the volume axis in Figure 4.2(3).1-4. As many of the individual demands (O(*)) as possible are satisfied in full by the available water. The remaining water is used to partially satisfy the demand of next highest priority, and any others are not satisfied at all.

4.2(3).1.1.2 Find the Outflow Demands which Correspond to a Specified Row in RCHTAB (subroutine DEMAND)

Purpose

DEMAND finds the individual and total outflow demands which apply at the end of the present interval for a specified level (row) in RCHTAB.

General Method

The approach is to determine the outflow demand for each active exit and accumulate them to find the total demand.

Evaluating the Demand for Exit N

The outflow demand for an individual exit consists of one or both of two components. Their presence or absence is indicated by two flags:

Component	Flag
fN(VOL)	ODFVFG(N)
gN(t)	ODGTFG(N)

Finding the fN(VOL) Component

If ODFVFG(N) is zero, there is no fN(VOL) component.

If ODFVFG(N) is greater than zero, there is a fN(VOL) component. The value of the flag is the column number in RCHTAB containing the value to be used to find the component:

```
    col  = ODFVFG(N)
    ODFV = fN(VOL) = (column value)*CONVF                              (12)
```

where CONVF is a conversion factor which can vary throughout the year. It is supplied by the user in the RCHRES Block of the User's Control Input. It can be used to incorporate effects into the simulation of, for example, seasonal variation in channel roughness.

If ODFVFG(N) is less than zero, there is an fN(VOL) component but the function fN is time varying. In this case the determination of the component is less direct. The absolute value of ODFVFG(N), say I, gives the element number of a vector COLIND() which contains a user supplied time series. The values in this time series indicate which pair of columns in RCHTAB are used to interpolate fN(VOL). For example, if COLIND(I) = 3.6 for a given time step, then the value is interpolated between those in columns 3 and 4:

```
ODFV = fN(VOL) = [0.6*(column4 value) + 0.4*(column3 value)]*CONVF     (13)
```

If the user has selected this option, he must supply the time series COLIND(I) in the EXT SOURCES Block of his UCI.

This method of outflow demand specification is useful where a set of rule curves (f(VOL)) are specified for releases from a reservoir, and it is necessary to move from one curve to another (gradually or suddenly) as time progresses in the simulation.

Finding the gN(VOL) Component

If ODGTFG(N) is zero, there is no gN(VOL) component. If ODGTFG(N) is greater than zero, there is a gN(t) component. The value of this flag is the element number of vector OUTDGT() which contains the required time series:

```
    FG2  = ODGTFG(N)
    ODGT = gN(t) = OUTDGT(FG2)                                         (14)
```

Combining the fN(VOL) and gN(t) Components

If an outflow demand has both of the components described above, the system expects the user to indicate which of the following options to use in combining them:

```
    1.  OD(N) = Min [fN(VOL),gN(t)]
    2.  OD(N) = Max [fN(VOL),gN(t)]
    3.  OD(N) = fN(VOL) + gN(t)                                        (15)
```

4.2(3).1.1.3 Solve Routing Equations used in Case 1. (subroutine SOLVE)

Purpose

SOLVE finds the point where Equations 4 and 8 intersect (case 1 in Figure 4.2(3).1-4).

General Approach

The general idea is to select a segment of Equation 8 and determine the point of intersection with Equation 4. If this point lies outside the selected segment, the code will select the adjacent segment (in the direction in which the point of intersection lies) and repeat the process. This continues until the point lies within the segment under consideration. To minimize searching, the segment in which the point of intersection was last located is used to start the process.

Solving the Simultaneous Linear Equations

Equations 4 and 9 can be written as:

A1*VOL+ B1*ROD = C1 (16)
A2*VOL+ B2*ROD = C2 (17)

These equations can be solved by evaluating the determinants:

$$DET = \begin{vmatrix} A1 & B1 \\ A2 & B2 \end{vmatrix} \quad DETV = \begin{vmatrix} C1 & B1 \\ C2 & B2 \end{vmatrix} \quad DETO = \begin{vmatrix} A1 & C1 \\ A2 & C2 \end{vmatrix} \tag{18}$$

In the code of this subroutine:

FACTA1 = A1 = 1.0/(COKS*DELTS) (19)
FACTA2 = A2 = ROD1 - ROD2 (20)
FACTB1 = B1 = 1.0 (21)
FACTB2 = B2 = V2 - V1 (22)
FACTC1 = C1 = OINT (23)
FACTC2 = C2 = (V2*ROD1) - (V1*ROD2) (24)

By substituting Equations 19 through 24 in Equation 18 the determinants are evaluated as:

DET = FACTA1*FACTB2 - FACTA2 (25)
DETV = OINT*FACTB2 - FACTC2 (26)
DETO = FACTA1*FACTC2 - FACTA2*OINT (27)

The coordinates of the point of intersection are:

VOL = DETV/DET (28)
RO = DETO/DET (29)

4.2(3).1.2 Calculate Outflows Without Using Hydraulic Routing (subroutine NOROUT)

Purpose

NOROUT is used to compute the rates and volumes of outflow from a RCHRES and the new volume in cases where no outflow demand has an f(VOL) component; that is, where all outflow demands are functions of time only.

Method

Equations 4 and 8 are illustrated for this situation in Figure 4.2(3).1-5. The solution procedure is similar to that used in subroutine ROUTE, except that: because no outflow demands depend on volume, no table look-up and interpolation is required to evaluate them, and the simultaneous solution of Equations 4 and 8 is easier.

The intercept of Equation 4 on the volume axis is found, as before, using Equation 10. If VOLINT is less than 0.0, there is no solution (case 3). The code takes similar action to that taken by subroutine ROUTE for this case.

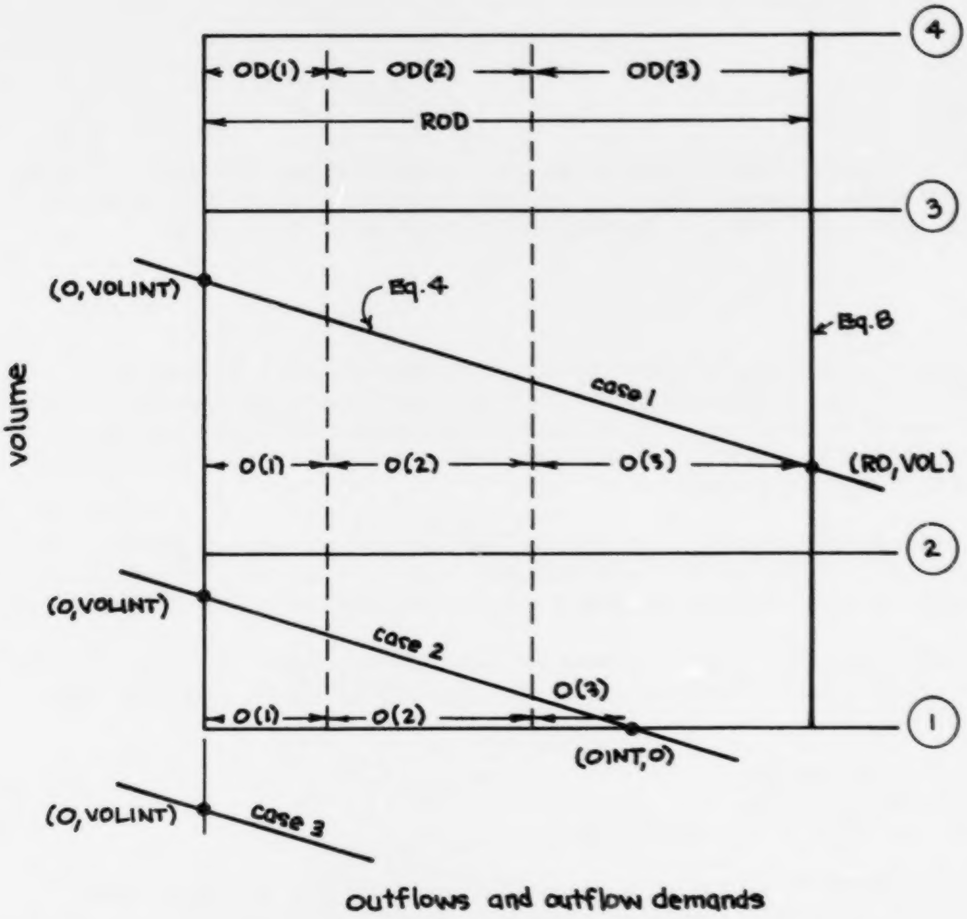
If VOLINT is greater than or equal to 0.0, the solution is either case 1 or case 2, as before. In either case, the first step is to evaluate the outflow demands:

```
    FG     = ODGTFG(N)
    OD(N)  = OUTDGT(FG)                                        (30)
    ROD    = OD(1) + ... OD(NEXITS)                            (31)
```

The intercept of Equation 4 on the volume axis (OINT) is found using Equation 11. If OINT is greater than ROD, Equations 4 and 8 can be solved (case 1):

```
    RO    = ROD
    O(*) = OD(*)                                               (32)
   And from Equations 4 and 10,
    VOL = VOLINT - COKS*RO*DELTS                               (33)
```

If OINT is less than or equal to ROD, Equations 4 and 8 cannot be solved (case 2). The physical meaning and the action taken by the code are identical to that described for subroutine ROUTE.



(–,–) are coordinates of points

② is row no. in RCHTAB which contains data for this level.

Figure 4.2(3).1-5 Graphical representation of the work performed by subroutine NOROUT

4.2(3).1.3 Compute Values of Auxiliary State Variables (subroutine AUXIL)

Purpose

AUXIL is used to compute the depth, stage, surface area, average depth and top width, given the volume of water in a RCHRES.

Method of Computing Depth

The basic problem is to interpolate a depth value between those given for discrete values of volume in RCHTAB. This raises the question of how the interpolation should be performed; for example, linear or quadratic. Whatever method is used, it should be consistent with the fact that volume is the integral of surface area with respect to depth.

Most RCHRES's are long and relatively narrow (Figure 4.2(3).1-6). To perform interpolation, it is assumed that surface area varies linearly with depth between neighboring levels (rows) in RCHTAB:

$$\mathrm{SAREA = SA1 + (SA2 - SA1)*RDEP} \quad (34)$$

where SAREA is the surface area at depth DEP; SA1, SA2 are the tabulated values of surface area immediately above and below SAREA; RDEP is the relative depth (DEP-DEP1)/(DEP2-DEP1); DEP1, DEP2 are the tabulated values of depth immediately above and below DEP.

By integrating the above equation with respect to depth and equating the result to volume:

$$\mathrm{(A*RDEP**2) + (B*RDEP) + C = 0.0} \quad (35)$$

where:
A = SA2 - SA1
B = 2.0*SA1
C = -(VOL - VOL1)/(VOL2 - VOL1)*(B + A)

Equation 35 provides a means of interpolating depth, given volume. There is a quadratic relation between RDEP and VOL. The equation can be solved for RDEP analytically but, in HSPF, Newton's method of successive approximations is used because it is probably faster in execution:

1. Calculation starts with an estimate of RDEP: RDEP1=0.5
2. The function FRDEP = (A*RDEP1**2) + (B*RDEP1) + C is evaluated
3. The derivative DFRDEP = 2.0*A*RDEP1 + B is evaluated
4. A new value RDEP2 = RDEP1 - FRDEP/DFRDEP is calculated
5. Steps 2-4 are repeated with RDEP1 = RDEP2 until the change in RDEP is small

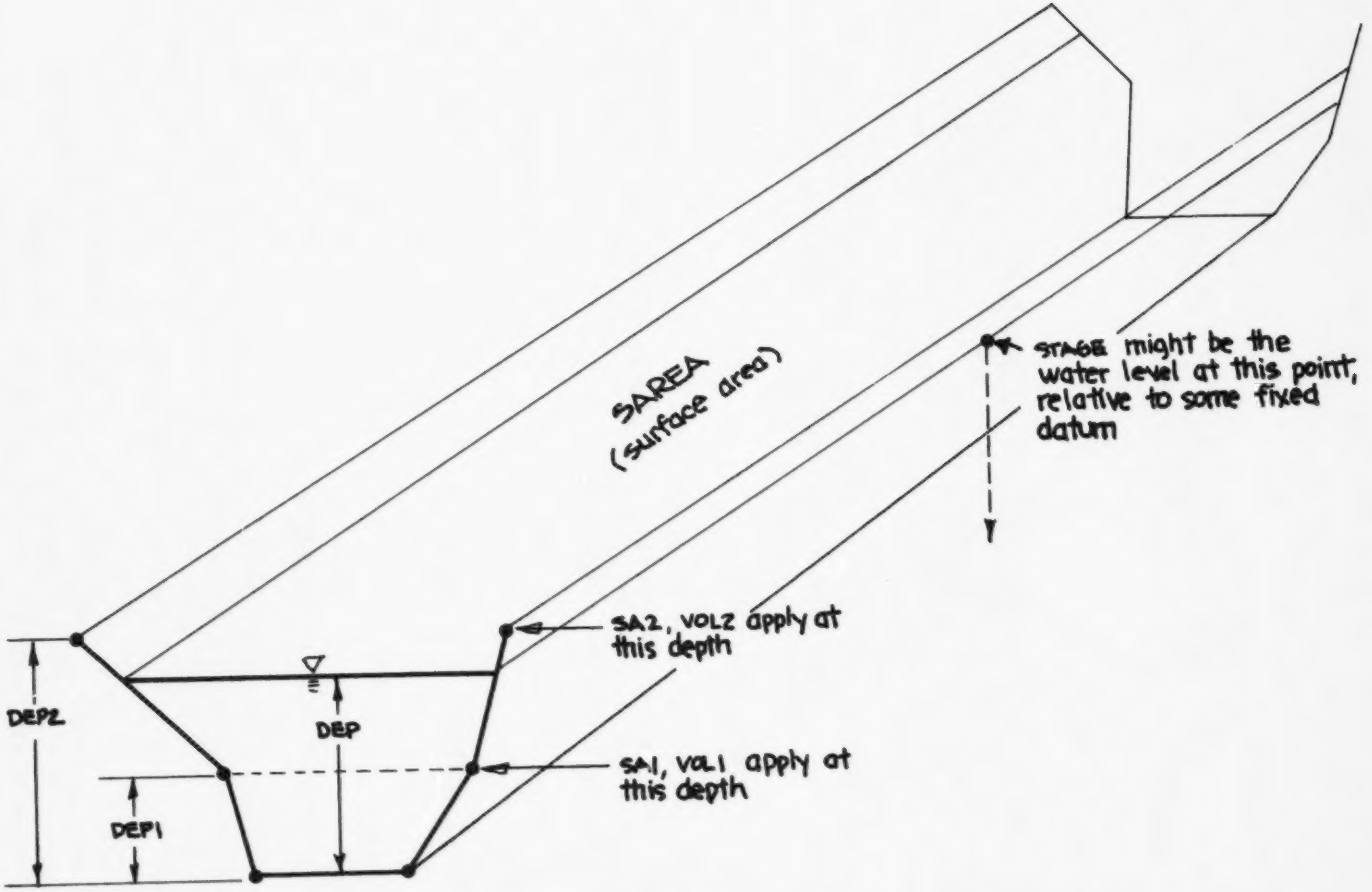

SAREA
(surface area)
STAGE might be the water level at this point, relative to some fixed datum
SA2, VOL2 apply at this depth
SA1, VOL1 apply at this depth
DEP2
DEP
DEP1

Figure 4:2(3).1-6 Illustration of quantities involved in calculation of depth

The depth is found using:

$$\mathrm{DEP} = \mathrm{DEP1} + \mathrm{RDEP2}*(\mathrm{DEP2} - \mathrm{DEP1}) \tag{36}$$

Computation of Other State Variables

STAGE is the name for any quantity which differs from DEP by a constant:

$$\mathrm{STAGE} = \mathrm{DEP} + \mathrm{STCOR} \tag{37}$$

where:
STCOR = the difference, supplied by the user

Surface area is computed using a formula based on Equation 34:

$$\mathrm{SAREA} = \mathrm{SA1} + \mathrm{A}*\mathrm{RDEP2} \tag{38}$$

Average depth is computed as:

$$\mathrm{AVDEP} = \mathrm{VOL}/\mathrm{SAREA} \tag{39}$$

The mean top width is found using:

$$\mathrm{TWID} = \mathrm{SAREA}/\mathrm{LEN} \tag{40}$$

where:
LEN = length of the RCHRES, supplied by the user

4.2(3).2 Prepare to Simulate Advection of Fully Entrained Constituents (Section ADCALC of Module RCHRES)

Purpose

ADCALC calculates values for variables which are necessary to simulate longitudinal advection of dissolved or entrained constituents. These variables are all dependent upon the volume and outflow values calculated in the hydraulics section (HYDR).

Approach

The outflow of an entrained constituent is a weighted mean of two quantities: one is an estimate based on conditions at the start of the time step, the other reflects conditions at the end of the step. The weighting factors are called JS and COJS (complement of JS) respectively. The values of the weighting coefficients depend on (1) the relative volume of stored water in the RCHRES compared to the volume leaving in a single time step and (2) the uniformity of the velocity across a cross section of the RCHRES. In order to represent these factors, two variables are defined: RAT and CRRAT. RAT is the ratio of RCHRES volume at the start of the interval to the outflow volume based on the outflow rate at the start of the interval:

$$RAT = VOLS/(ROS*DELTS) \tag{1}$$

where:

VOLS = volume of water at the start of interval in ft3 or m3
ROS = outflow rate at start of interval in ft3/s or m3/s
DELTS = number of seconds in interval

The parameter CRRAT is defined as the ratio of maximum velocity to mean velocity in the RCHRES cross section under typical flow conditions. CRRAT must always have a value of 1.0 or greater. A value of 1.0 corresponds to a totally uniform velocity (plug flow) across the RCHRES.

Determination of JS and COJS

If the value of RAT is greater than that of CRRAT, it is assumed that all outflow over a given time interval was contained in the RCHRES at the start of the interval, and the mean rate of outflow of material is wholly dependent upon the rate of outflow at the start of the interval (JS = 1.0). If the value of RAT is less than CRRAT, it is assumed that part of the water in the outflow entered the RCHRES as inflow during the same interval; in this case, the concentration of inflowing material will affect the outflow concentration in the same interval, and JS will have a value less than 1.0. The relationship of RAT, CRRAT, and JS is illustrated in Figure 4.2(3).2-1. COJS is (1.0 - JS).

Another way to interpret the relationship of these variables is that no inflowing material is present in the outflow in the same interval if the outflow volume is less than (VOLS/CRRAT).

Calculation of Components of Outflow Volume

Components of outflow volume based on conditions at the start of the interval (SROVOL) and the end of the interval (EROVOL) are calculated as:

SROVOL = JS*ROS*DELTS (2)
EROVOL = COJS*RO*DELTS

where:
SROVOL = outflow volume component based on start of interval, in ft3/interval or m3/interval
EROVOL = outflow volume component based on end of interval, in ft3/interval or m3/interval
ROS = outflow rate at start of interval, in ft3/s or m3/s
RO = outflow rate at end of interval, in ft3/s or m3/s
DELTS = number of seconds in interval

Likewise, if there is more than one exit gate for the RCHRES, the corresponding outflow components for each unit, based on conditions at the start and end of each interval, are calculated as:

SOVOL(N) = JS*OS(N)*DELTS (3)
EOVOL(N) = COJS*O(N)*DELTS

where:
SOVOL(N) = outflow volume component based on start of interval for exit gate N, in ft3/interval or m3/interval
EOVOL(N) = outflow volume component based on end of interval for exit gate N, in ft3/interval or m3/interval
OS(N) = outflow rate at start of interval for exit gate N, in ft3/s or m3/s
O(N) = outflow rate at end of interval for exit gate N, in ft3/s or m3/s
DELTS = number of seconds in interval

It should be noted that SROVOL, EROVOL, SOVOL(N), and EOVOL(N) are not actual outflows from the RCHRES, but instead are components of outflow based on conditions at the start or end of the interval. These variables are used in subroutine ADVECT to estimate the advection of constituents.

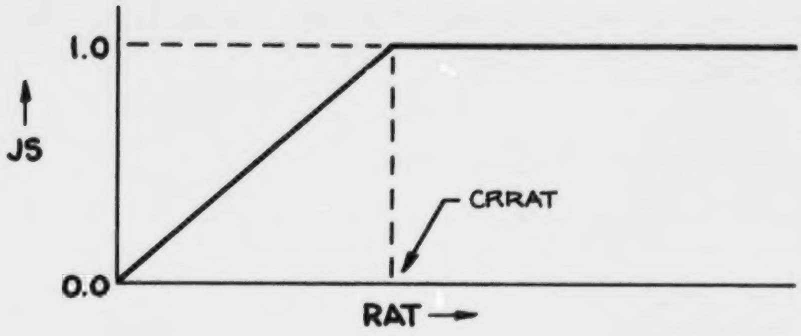


Figure 4.2(3).2-1 Determination of weighting factors for advection calculations

4.2(3).3 Simulate Conservative Constituents (Section CONS of Module RCHRES)

Purpose

CONS simulates constituents which, for all practical purposes, do not decay with time or leave the RCHRES by any mechanism other than advection. Examples include:

total dissolved solids
chlorides
pesticides and herbicides which decay very slowly

Figure 4.2(3).3-1 illustrates the fluxes of conservative material which are modeled in section CONS.

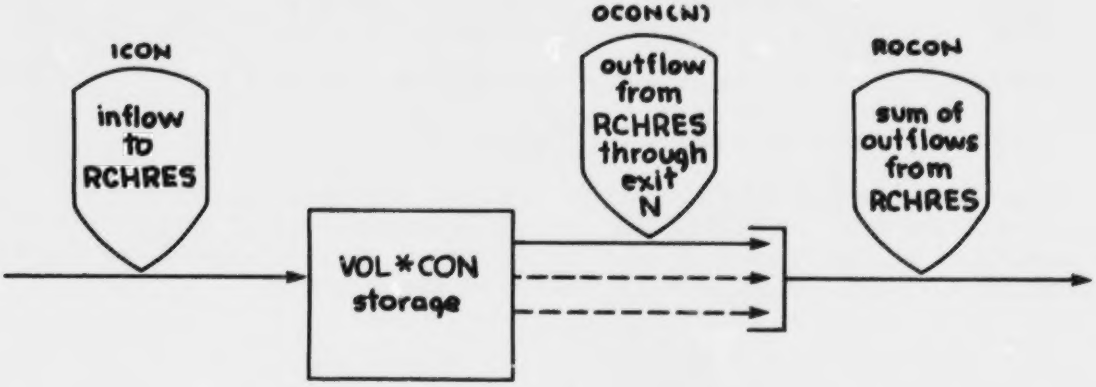

ICON
inflow to RCHRES
VOL*CON storage
OCON(N)
outflow from RCHRES through exit N
ROCON
sum of outflows from RCHRES

Figure 4.2(3).3-1 Flow diagram for conservative constituents in the CONS section of the RCHRES Application Module

Method

Subroutine CONS performs only three functions. First, a value for inflow of material (ICON) is obtained and converted to internal units. Next, CONS calls subroutine ADVECT to perform longitudinal advection of this material and the material already contained in the RCHRES. Finally, CONS calculates the mass of material remaining in the RCHRES after advection; this value, RCON, is necessary for the mass balance checks on conservatives and is calculated as:

$$RCON = CON*VOL \qquad (1)$$

where:

RCON = mass of material in RCHRES after advection
CON = concentration of conservative after advection
VOL = volume of water in RCHRES at end of interval in ft3 or m3

Additional Requirements

HSPF allows a maximum of ten conservative constituents. The user selects the units for each constituent; thus, different conservative constituents may have different units. However, in order to provide this flexibility, additional input is required. For each constituent the following information must be provided in the User's Control Input:

1. CONID: the name of the constituent (up to 20 characters long)

2. QTYID: this string (up to 8 characters) contains the units used to describe the quantity of constituent entering or leaving the RCHRES, or the total quantity of material stored in it. Examples of possible units for QTYID are 'kg' for kilograms or 'lbs' for pounds

3. CONCID: the concentration units for each conservative (up to 8 characters long); examples are 'mg/l' or 'lbs/ft3'

4. CONV: conversion factor from QTYID/VOL to desired concentration units: CONC = CONV*(QTY/VOL) (in English system, VOL is expressed in ft3) (in metric system, VOL is expressed in m3)
   For example, if:
       CONCID is mg/l
       QTYID is kg
       VOL is in m3
   then CONV = 1000.0

4.2(3).3.1 Simulate Advection of Constituent Totally Entrained in Water (subroutine ADVECT)

Purpose

ADVECT computes the concentration of material in a RCHRES and the quantities of material that leave the RCHRES due to longitudinal advection through active exit gates. ADVECT is a generalized subroutine, and is called by each module section which simulates constituents which undergo normal longitudinal advection.

Assumptions

Two assumptions are made in the solution technique for normal advection:

1. Each constituent advected by calling subroutine ADVECT is uniformly dispersed throughout the waters of the RCHRES.
2. Each constituent is completely entrained by the flow; that is, the material moves at the same horizontal velocity as the water.

Method

The equation of continuity may be written as:

$$IMAT - ROMAT = (CONC*VOL) - (CONCS*VOLS) \qquad (2)$$

where:

IMAT = inflow of material over the interval
ROMAT = total outflow of material over the interval
CONCS and CONC = concentrations at the start and end of the interval
VOLS and VOL = volume of water stored in the RCHRES at the start and end of the interval (m3 or ft3)

The other basic equation states that the total outflow of material over the time interval is a weighted mean of two estimates; one based on conditions at the start of the interval, the other on ending conditions:

$$ROMAT = ((JS*ROS*CONCS) + (COJS*RO*CONC))*DELTS \qquad (3)$$

where:

JS = weighting factor and COJS = 1.0 - JS
ROS and RO = rates of outflow at the start and end of the interval (m3/s or ft3/s)
DELTS = interval, in seconds

Using equations (2) in Section 4.2(3).2 (Subroutine ADCALC), equation (3) can be written:

$$ROMAT = (SROVOL*CONCS) + (EROVOL*CONC) \quad (4)$$

where SROVOL and EROVOL are as defined earlier.

By combining equations (2) and (4) we can solve for CONC:

$$CONC = (IMAT + CONCS*(VOLS - SROVOL))/(VOL + EROVOL) \quad (5)$$

The total amount of material leaving the RCHRES during the interval is calculated from equation (4).

If there is more than one active exit from the RCHRES, the amount of material leaving through each exit gate is calculated as:

$$OMAT = SOVOL*CONCS + EOVOL*CONC \quad (6)$$

where:

OMAT = amount of material leaving RCHRES through individual exit gate
SOVOL = outflow volume component for individual exit gate based on start of interval
EOVOL = outflow volume component for individual exit gate based on end of interval

(SOVOL and EOVOL are defined in Section 4.2(3).2)

If the RCHRES goes dry during the interval, the concentration at the end of the interval is undefined. The total amount of material leaving the RCHRES is:

$$ROMAT = IMAT + (CONCS*VOLS) \quad (7)$$

If there is more than one active exit from the RCHRES, the amount of material leaving through each exit gate from a RCHRES which has gone dry during the interval is calculated as:

$$OMAT = (SOVOL/SROVOL)*ROMAT \quad (8)$$

The units in the foregoing equations are:

VOLS,VOL	m3 or ft3 (call these volunits)
SROVOL,etc	volunits/interval
CONCS,CONC	user defined (call these concunits)
IMAT,ROMAT,etc	concunits*volunits/interval

4.2(3).4 Simulate Heat Exchange and Water Temperature (Section HTRCH of Module RCHRES)

Purpose

The purpose of this code is to simulate the processes which determine the water temperature in a reach or mixed reservoir. Water temperature is one of the most fundamental indices used to determine the nature of an aquatic environment. Most processes of functional importance to an environment are affected by temperature. For example, the saturation level of dissolved oxygen varies inversely with temperature. The decay of reduced organic matter, and hence oxygen demand caused by the decay, increases with increasing temperature. Some form of temperature dependence is present in nearly all processes. The prevalence of individual phytoplankton and zooplankton species is often temperature dependent.

Required Time Series

Five time series of meteorological data are required to simulate the temperature balance within a RCHRES. These are:

1. solar radiation in langleys/interval
2. cloud cover expressed as tenths
3. air temperature in degrees C
4. dewpoint temperature in degrees C
5. wind speed in meters/interval

Note that solar radiation data are usually available as daily totals. The user must convert these data to, say, hourly or two hourly values before using them in HSPF. If the standard HSPF disaggregation rule were used, a daily value would be divided into equal increments for each interval of the day; this would not account for the rising and setting of the sun! A similar kind of preprocessing needs to be done if daily max/min air temperatures are used.

Schematic View of Fluxes and Storages

Figure 4.2(3).4-1 illustrates the fluxes involved in this module section. There are no significant internal sources or sinks of temperature within a RCHRES. Changes in heat content are due only to transport processes across the RCHRES boundaries. Module section HTRCH considers two major processes: heat transfer by advection, and heat transfer across the air-water interface. The processes of diffusion and dispersion are not considered in HSPF.

Heat transfer by advection is simulated by treating water temperature as a thermal concentration. This enables the use of subroutine ADVECT, a

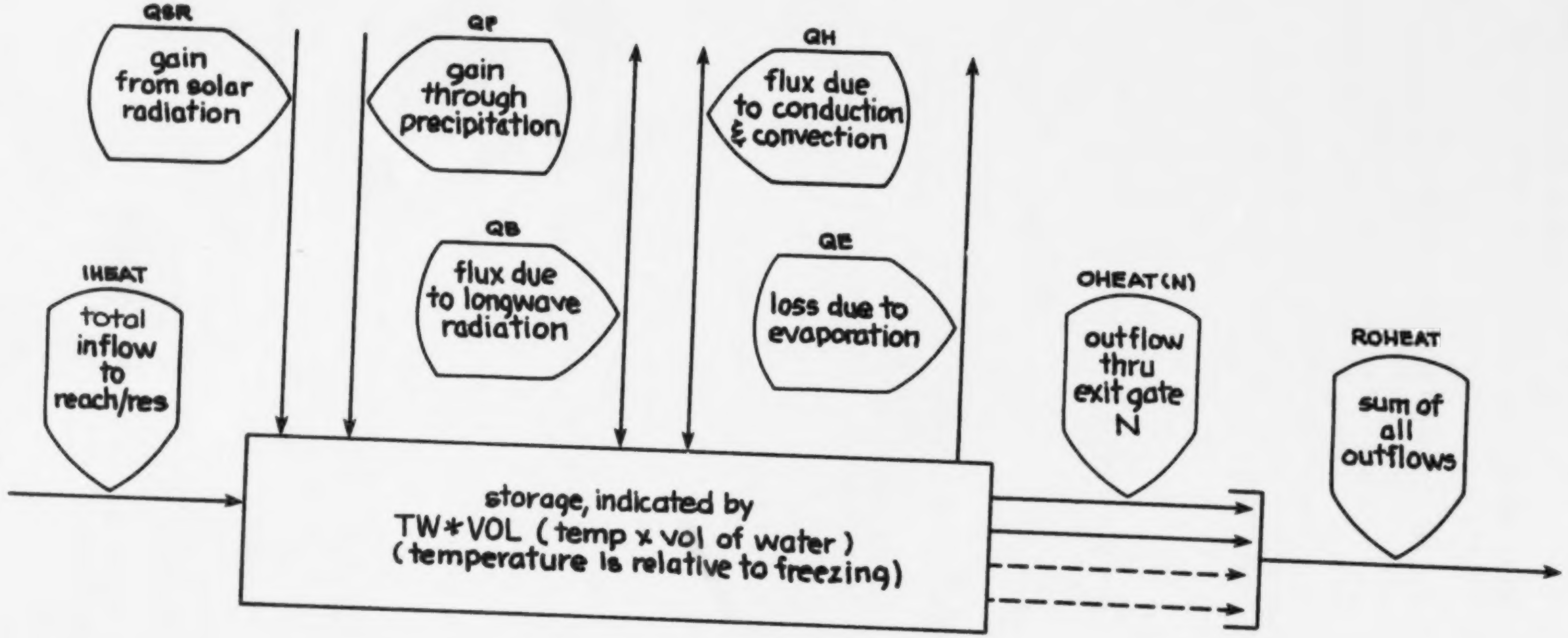

QSR
gain from solar radiation
QP
gain through precipitation
QH
flux due to conduction & convection
QB
flux due to longwave radiation
QE
loss due to evaporation
IHEAT
total inflow to reach/res
OHEAT (N)
outflow thru exit gate N
ROHEAT
sum of all outflows
storage, indicated by
TW*VOL (temp x vol of water)
(temperature is relative to freezing)

Figure 4.2(3).4-1 Flow diagram for HTRCH Section of the RCHRES Application Module standard subroutine which calculates advective transport of constituents totally entrained in the moving water.

Heat is transported across the air-water interface by a number of mechanisms, and each must be evaluated individually. The net transport across the air-water interface is the sum of the individual effects. Mechanisms which can increase the heat content of the water are absorption of solar radiation, absorption of longwave radiation, and conduction-convection. Mechanisms which decrease the heat content are emission of longwave radiation, conduction-convection, and evaporation.

Shortwave Solar Radiation

The shortwave radiation absorbed by a RCHRES is approximated by the following equation:

$$QSR = 0.97*CFSAEX*SOLRAD*10.0 \qquad (1)$$

where:

QSR = shortwave radiation in kcal/m2.interval
0.97 = fraction of incident radiation which is assumed absorbed (3 percent is assumed reflected)
CFSAEX = ratio of radiation incident to water surface to radiation incident to gage where data were collected. This factor also accounts for shading of the water body, eg. by trees
SOLRAD = solar radiation in langleys/interval
10.0 = conversion factor from langleys to kcal/m2

Longwave Radiation

All terrestial surfaces, as well as the atmosphere, emit longwave radiation. The rate at which each source emits longwave radiation is dependent upon its temperature. The longwave radiation exchange between the atmosphere and the RCHRES is estimated using the formula:

$$QB = SIGMA*((TWKELV**4) - KATRAD*(10**-6)*CLDFAC*(TAKELV**6))*DELT60 \qquad (2)$$

where:

QB = net transport of longwave radiation in kcal/m2.interval
SIGMA = Stephan-Boltzman constant multiplied by 0.97 to account for emissivity of water
TWKELV = water temperature in degrees Kelvin
KATRAD = atmospheric longwave radiation coefficient with a typical value of 9.0
CLDFAC = 1.0 + (.0017*C**2)
TAKELV = air temperature in degrees Kelvin corrected for elevation difference
C = cloud cover, expressed as tenths (range 0 through 10)
DELT60 = DELT(mins) divided by 60

Both atmospheric radiation to the water body and back radiation from the water body to the atmosphere are considered in this equation. QB is positive for transport of entergy from the water body to the atmosphere.

Conduction-Convection

Conductive-convective transport of heat is caused by temperature differences between the air and water. Heat is transported from the warmer medium to the cooler medium; heat can therefore enter or leave a water body, depending upon its temperature relative to air temperature. HSPF assumes that the heat transport is proportional to the temperature difference between the two media. The equation used is:

$$QH = CFPRES*(KCOND*10**-4)*WIND*(TW - AIRTMP) \qquad (3)$$

where:

QH = conductive-convective heat transport in kcal/m2.interval
CFPRES = pressure correction factor dependent on elevation
KCOND = conductive-convective heat transport coefficient (typically in the range of 1 to 20)
WIND = windspeed in m/interval
TW = water temperature in degrees C
AIRTMP = air temperature in degrees C

QH is positive for heat transfer from the water to the air.

Evaporative Heat Loss

Evaporative heat transport occurs when water evaporates from the water surface. The amount of heat lost depends on the latent heat of vaporization for water and on the quantity of water evaporated. For purposes of water temperature simulation, HSPF uses the following equation to calculate the amount of water evaporated:

$$EVAP = (KEVAP*10**-9)*WIND*(VPRESW - VPRESA) \qquad (4)$$

where:

EVAP = quantity of water evaporated in m/interval
KEVAP = evaporation coefficient with a typical value of 1 to 5
WIND = wind movement 2 m above the water surface in m/interval
VPRESW = saturation vapor pressure at the water surface in mbar
VPRESA = vapor pressure of air above water surface in mbar

The heat removed by evaporation is then calculated:

$$QE = HFACT*EVAP \qquad (5)$$

where:
QE = heat loss due to evaporation in kcal/m2.interval
HFACT = heat loss conversion factor (latent heat of vaporization multiplied by density of water)

Heat Content of Precipitation

In module section HYDR an option exists to include the input of water from precipitation falling directly on the water surface. If this option is activated, it is necessary to assign a temperature to the water added to the RCHRES in this manner. HSPF assumes that precipitation has the same temperature as the water surface on which it falls.

Net heat exchange

The net heat exchange at the water surface is represented as:

$$QT = QSR - QB - QH - QE + QP \qquad (6)$$

where:
QT = net heat exchange at water surface in kcal/m2.interval
QSR = net heat transport from incident shortwave radiation
QB = net heat transport from longwave radiation
QH = heat transport from conduction-convection
QE = heat transport from evaporation
QP = heat content of precipitation

Calculation of Water Temperature

Of the five heat transport mechanisms across the air-water interface, three are significant and dependent upon water temperature. In order to obtain a stable solution for water temperature, these three terms (QB, QH, QE) are evaluated for the temperature at both the start and end of the interval, and the average of the two values is taken (trapezoidal approximation). For this purpose, the unknown ending temperature is approximated by performing a Taylor series expansion about the starting temperature, and ignoring nonlinear terms. This formulation leads to the following equation for the change in water temperature over the interval:

$$DELTTW = CVQT*QT/(1.0 + SPD*CVQT) \qquad (7)$$

where:
DELTTW = change in water temperature in degrees C
CVQT = conversion factor to convert total heat exchange expressed in kcal/m2.interval to degrees C/interval (volume dependent)
QT = net heat exchange in kcal/m2.interval (with terms evaluated at starting temperature)
SPD = sum of partial derivatives of QB, QH, and QE with respect to water temperature

The heat exchange calculations do not give realistic results when the water body becomes excessively shallow. Consequently, heat transport across the air-water interface is not considered if the average depth of water in the RCHRES falls below 2 in.

4.2(3).4.1 Correct Air Temperature for Elevation Difference (subroutine RATEMP)

Purpose

The purpose of this code is to correct air temperature for any elevation difference between the RCHRES and the temperature gage.

Approach

The lapse rate for air temperature is dependent upon whether or not precipitation occurs during the time interval. If precipitation does occur, a wet lapse rate of 1.94E-3 degrees C/ft is assumed. Otherwise, a dry lapse rate which is a function of time of day is used. A table of 24 hourly dry lapse rates is built into the HSPF system. The corrected air temperature is:

$$\mathrm{AIRTMP} = \mathrm{GATMP} - \mathrm{LAPS}*\mathrm{ELDAT} \qquad (8)$$

where:
AIRTMP = corrected air temperature in degrees C
GATMP = air temperature at gage
LAPS = lapse rate in degrees C/ft
ELDAT = elevation difference between mean RCHRES elevation and gage elevation in feet (ELDAT is positive if mean RCHRES elevation is greater than gage elevation)

4.2(3).5 Simulate Inorganic Sediment (Section SED of Module RCHRES)

Purpose

The purpose of this code is to simulate the transport and deposition of inorganic sediment in free-flowing reaches and mixed reservoirs. The modeling of sediment in channels may be needed for analysis of such problems as:

1. Structural instability of bridge piers or water intakes caused by scouring.

2. Reduction of reservoir capacity and clogging of irrigation canals and navigable waterways due to deposition.

3. Reduction of light available to aquatic organisms caused by suspended sediment.

4. Transport of adsorbed pollutants such as fertilizers, herbicides, and pesticides.

Schematic View of Fluxes and Storages

Figures 4.2(3).5-1 and 4.2(3).5-2 show the principal state variables and fluxes with which subroutine SED deals.

For modeling purposes, HSPF divides the inorganic sediment load into two components, washload and sandload, each with its own properties. Washload consists of clay and fine silt, and is modeled as a conservative substance, apart from deposition on the bed, controlled by a user-specified "sinking rate". However, sinking is significant only in reservoirs or slow moving reaches. Once washload material has settled out of a RCHRES, HSPF assumes that it cannot be resuspended. Thus, the only source of washload modeled is erosion from the land surface. Sandload consists of sand and gravel from either the land surface or the channel system. The scour or deposition of sandload is assumed not to affect the hydraulic properties of the channel.

The PERLND module of HSPF simulates removal of total inorganic sediment due to washoff from the land surface and erosion from gullies. The model user must divide it into the two components (washload and sandload) so that this material can be routed through the channel system by the RCHRES module.

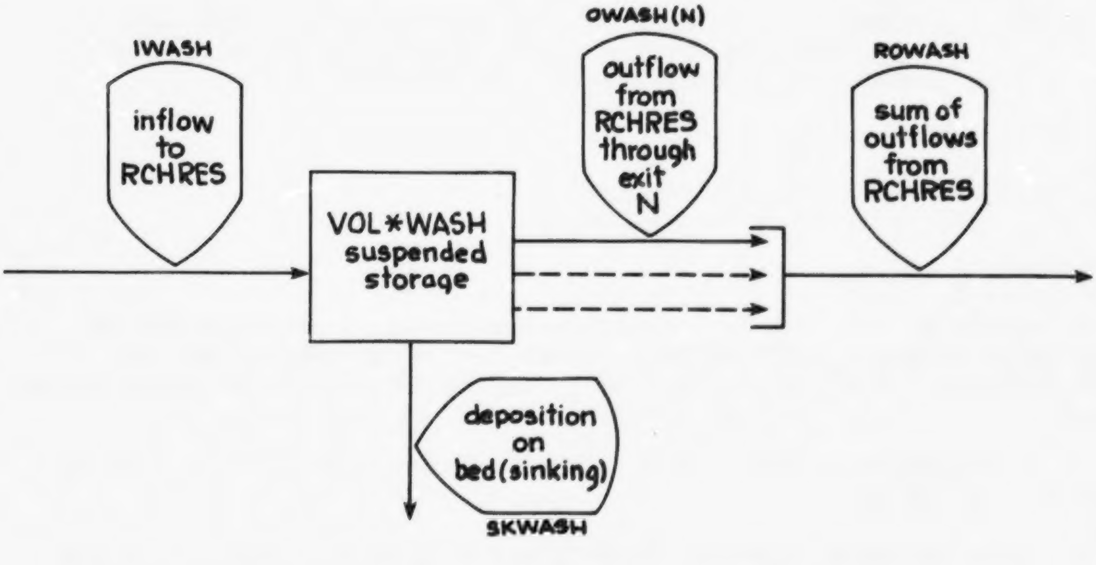

IWASH
inflow to RCHRES
OWASH(N)
outflow from RCHRES through exit N
ROWASH
sum of outflows from RCHRES
VOL*WASH suspended storage
deposition on bed(sinking)
SKWASH

Figure 4.2(3).5-1 Flow diagram for washload in the SED section of the RCHRES Application Module

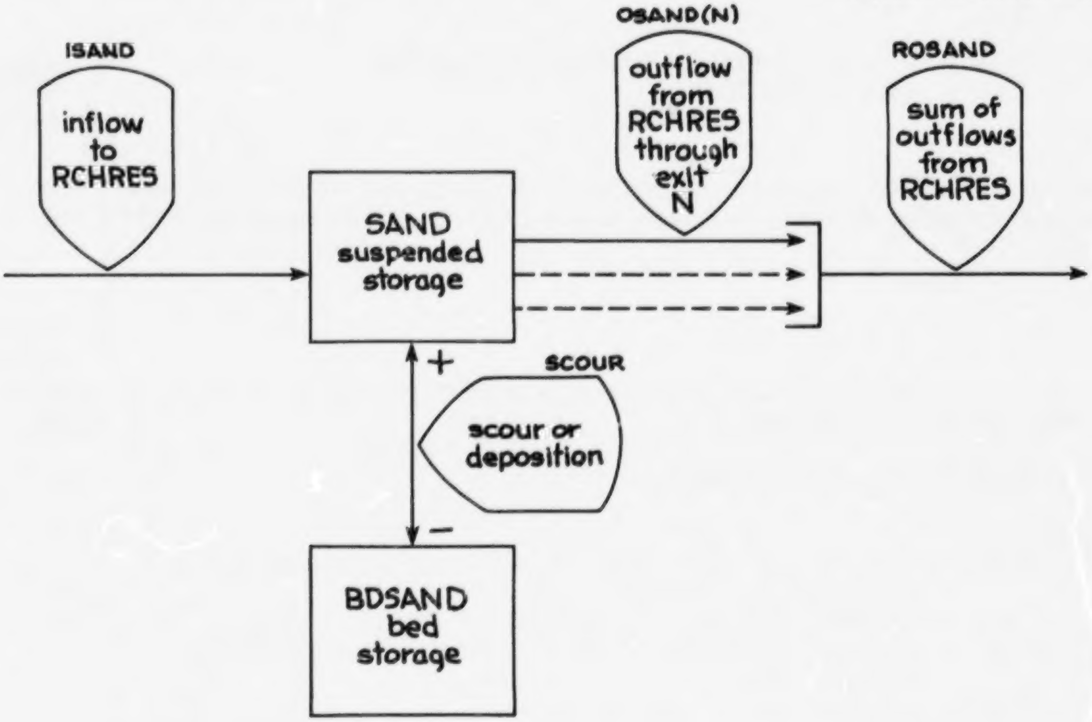


Figure 4.2(3).5-2 Flow diagram for sandload in the SED Section of the RCHRES Application Module

4.2(3).5.1 Simulate Behavior of Washload (Clays and Fine Silt) (subroutine WASHLD)

Purpose

WASHLD simulates the transport and settling of washload materials.

Procedure

The modeling effort consists of two steps. First, subroutine ADVECT is called to perform advective transport. Then the quantity of washload which settles out of the water is calculated and the concentration of washload remaining in the RCHRES is determined, by calling subroutine SINK.

4.2(3).5.1.1 Simulate Sinking of Suspended Material (subroutine SINK)

Purpose

SINK calculates the quantity of material settling out of a RCHRES and determines the resultant change in concentration of the material within the RCHRES.

Method

The portion of material settling out of a RCHRES during an interval is calculated by the equation:

SNKOUT = CONC*(KSET/AVDEPE) (1)

where:
SNKOUT = fraction of material which settles out (reduction of concentration/interval)
CONC = concentration of material before deposition
KSET = sinking rate in ft/interval (dependent upon RCHRES characteristics and type of material)
AVDEPE = average depth of water expressed in feet

In any interval in which KSET is greater than AVDEPE, all the material in the RCHRES sinks out of the water.

The mass of material sinking out of the RCHRES is calculated as:

SNKMAT = SNKOUT*VOL (2)

where:
SNKMAT = mass of material that settles out during the interval expressed as mass.ft3/l.interval or mass.m3/l.interval
VOL = volume of water in RCHRES in ft3 or m3

4.2(3).5.2 Simulate Behavior of Sand/gravel (subroutine SANDLD)

Purpose

SANDLD simulates the transport and deposition of the sandload component of sediment

Approach

There are two storages of sand/gravel material considered in subroutine SANDLD: SAND, which is the sandload suspended in the RCHRES water, and

BDSAND, which is the mass of sandload stored on the bed of the RCHRES. Exchange of sandload material between these two storages is dependent upon the velocity of the water in the RCHRES. The potential load of sand which may remain in suspension is calculated from the equation:

PSAND = KSAND*(AVVELE**EXPSND) (3)

where:
PSAND = potential sandload expressed in mg/l
KSAND = coefficient to sandload suspension equation
AVVELE = average velocity in ft/sec
EXPSND = exponent to sandload suspension equation

The potential outflow of sand during the interval is:

PROSND = (SANDS*SROVOL) + (PSAND*EROVOL) (4)

where:
PROSND = potential sand outflow
SANDS = concentration of sand at start of interval (mg/l)
SROVOL and EROVOL are as defined in Section 4.2(3).2

The potential scour from, or deposition to, the bed storage is found using the continuity equation:

PSCOUR = (VOL*PSAND) - (VOLS*SANDS) + PROSND - ISAND (5)

where:
PSCOUR = potential scour (+) or deposition (-)
VOL = volume of water in RCHRES at the end of the interval (ft3 or m3)
VOLS = volume of water in RCHRES at the start of interval (ft3 or m3)
ISAND = total inflow of sand into RCHRES during interval

The terms in equations (4) and (5) have the units of concentration (mg/l) * volume (ft3 or m3) / interval.

The potential scour is compared to the amount of sand material on the bottom surface available for resuspension. If scour demand is less than available bottom sands, the demand is satisfied in full, and the bed storage is adjusted accordingly. The new suspended concentration is PSAND. If the potential scour cannot be satisfied by bed storage, all of the available bedsand is suspended, and bed storage is exhausted. The concentration of suspended sandload is calculated as:

SAND = (ISAND + SCOUR + SANDS*(VOLS - SROVOL))/(VOL + EROVOL) (6)

where:
SAND = concentration of sand at end of interval
SCOUR = sand scoured from, or deposited to, the bottom
SANDS = concentration of sand at start of interval

The total amount of sand leaving the RCHRES during the interval is:

$$\mathrm{ROSAND} = \mathrm{SROVOL}*\mathrm{SANDS} + \mathrm{EROVOL}*\mathrm{SAND} \tag{7}$$

If a RCHRES goes dry during an interval, all the sand in suspension at the beginning of the interval is assumed to settle out, and the bed storage is correspondingly increased.

4.2(3).6 Simulate First Order Decay Constituents (Section FSTORD of Module RCHRES)

Purpose

The purpose of this code is to enable the user to model first order decay behavior for constituents not involved in any other reactions in the RCHRES module. A first order reaction is one in which the rate of reaction is proportional to the concentration of the substance reacting. Observed data for natural waters indicate most reactions exhibit first order kinetics (Thomann 1972). In HSPF, first order kinetics are used to represent a number of processes such as BOD decay, reaeration, and nitrification; in such cases, equations for the reaction kinetics are contained in the subroutines used for simulation of those constituents. The FSTORD section however, is designed to handle constituents not modeled elsewhere in the RCHRES module, but which exhibit degradation dependent upon concentration of constituent and amount of elapsed time. Examples of constituents which may be modeled in this way are:

1. bacteria (total coliforms, fecal coliforms, fecal streptococci)
2. pesticides, herbicides, or other rapidly degrading chemicals
3. radioactive materials

Schematic View of Fluxes and Storage

Figure 4.2(3).6-1 shows the fluxes of first order material which are modeled in section FSTORD.

Approach

The general equation for first order decay is:

$$\mathrm{DECAY} = \mathrm{DECFR}*\mathrm{CONC} \tag{1}$$

where:

DECAY = amount of decay during interval, expressed as a reduction in concentration/interval
DECFR = first order decay coefficient expressed as 1/time
CONC = concentration of constituent at start of interval

This equation gives an exact representation of radioactive decay. However, most processes occurring in natural waters are dependent not only on elapsed time, but also on water temperature. Consequently, the equation for first order decay used in section FSTORD contains a temperature correction. The following equation is used to determine the fraction of a constituent which decays during the simulation time interval:

```
     DECFR = (KFST20*TCFST**(TW - 20.))*DELT60                                  (2)
```

where:

KFST20 = decay coefficient at 20 degrees C (1/hr)
TCFST = temperature correction coefficient for first order decay
TW = water temperature in degrees C
DELT60 = simulation interval in minutes divided by 60; this factor adjusts hourly decay rate to the simulation interval.

The amount of constituent remaining at the end of the time step is calculated as:

```
     FSTOR = (1.0 - DECFR)*FSTORS                                               (3)
```

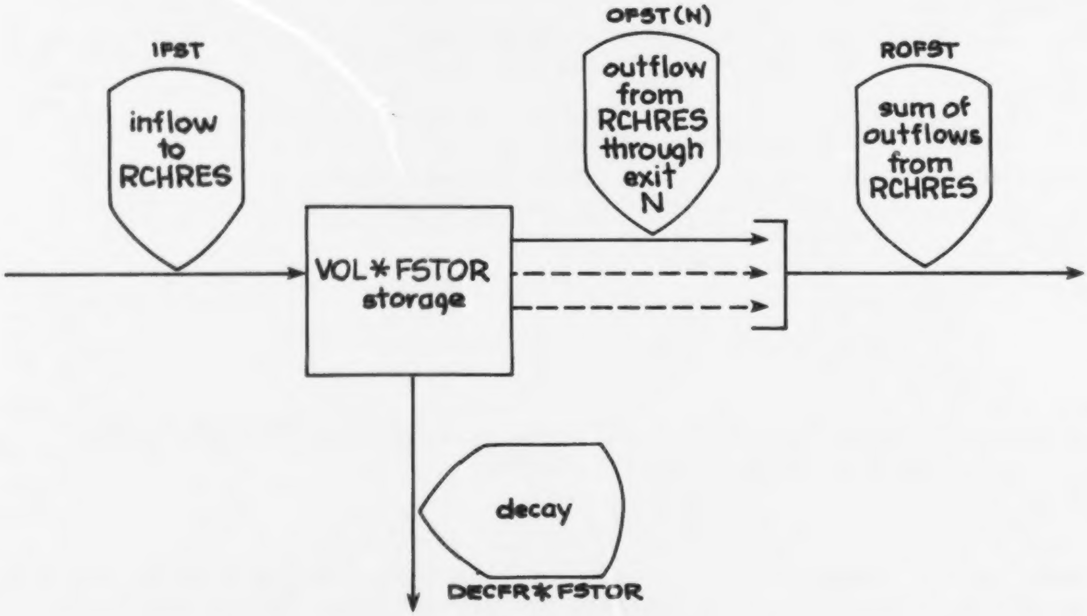


Figure 4.2(3).6-1 Flow diagram for first order decay in the FSTORD section of the RCHRES Application module

where:

FSTOR = concentration of constituent at end of interval
DECFR = fraction of constituent which decays during interval
FSTORS = concentration of constituent after advective transport, but prior to decay calculation

Additional Requirements

HSPF allows a maximum of ten first order constituents. The user selects the units for each constituent; thus, different first order constituents may have different units. For example, the user may simulate fecal and total coliforms expressed in organisms per 100 ml and a pesticide expressed in milligrams per liter in the same simulation. In order to provide this flexibility, additional input is required. For each constituent the following information must be provided in the User's Control Input:

1. FSTID: the name of the constituent (up to 20 characters long)

2. QTYID: this string (up to 8 characters) contains the units used to describe the quantity of constituent entering or leaving the RCHRES, and the total quantity of material stored in it. Examples of possible units for QTYID are 'Morg' for millions of organisms or 'lbs' for pounds

3. CONCID: the concentration units for each decay constituent (up to 8 characters long); examples are '#/100ml' or 'mg/l'

4. CONV: conversion factor from QTYID/VOL to desired concentration units: CONC = CONV*(QTY/VOL)
   (in English system, VOL is expressed in ft3)
   (in metric system, VOL is expressed in m3)
   For example, if
     CONCID is mg/l
     QTYID is kg
     VOL is m3
   then CONV = 1000.0

4.2(3).7 Simulate Constituents Involved in Biochemical Transformations (Section RQUAL of Module RCHRES)

RQUAL is the parent subroutine to the four subroutine groups which simulate constituents involved in biochemical transformations. Within module section RQUAL the following constituents may be simulated:

dissolved oxygen
biochemical oxygen demand
ammonia

nitrite
nitrate
orthophosphorus
phytoplankton
benthic algae
zooplankton
dead refractory organic nitrogen
dead refractory organic phosphorus
dead refractory organic carbon
total inorganic carbon
pH
carbon dioxide

Four additional quantities are estimated from simulation of these constituents. These quantities are total organic nitrogen, total organic phosphorus, total organic carbon, and potential biochemical oxygen demand. The definition of these quantities is determined by their method of calculation:

$$\mathrm{TORN} = \mathrm{ORN} + \mathrm{CVBN}*(\mathrm{ZOO} + \mathrm{PHYTO} + \mathrm{BOD}/\mathrm{CVBO}) \tag{1}$$

$$\mathrm{TORP} = \mathrm{ORP} + \mathrm{CVBP}*(\mathrm{ZOO} + \mathrm{PHYTO} + \mathrm{BOD}/\mathrm{CVBO}) \tag{2}$$

$$\mathrm{TORC} = \mathrm{ORC} + \mathrm{CVBC}*(\mathrm{ZOO} + \mathrm{PHYTO} + \mathrm{BOD}/\mathrm{CVBO}) \tag{3}$$

$$\mathrm{POTBOD} = \mathrm{BOD} + \mathrm{CVNRBO}*(\mathrm{ZOO} + \mathrm{PHYTO}) \tag{4}$$

where:
TORN = total organic nitrogen in mg N/l
TORP = total organic phosphorus in mg P/l
TORC = total organic carbon in mg C/l
POTBOD = potential BOD in mg O/l
ORN = dead refractory organic nitrogen in mg N/l
ORP = dead refractory organic phosphorus in mg P/l
ORC = dead refractory organic carbon in mg C/l
BOD = biochemical oxygen demand from dead nonrefractory organic materials in mg O/l
CVBN = conversion from mg biomass to mg nitrogen
CVBP = conversion from mg biomass to mg phosphorus
CVBC = conversion from mg biomass to mg carbon
CVNRBO = conversion from mg biomass to mg biochemical oxygen demand (with allowance for non-refractory fraction)
CVBO = conversion from mg biomass to mg oxygen
ZOO = zooplankton in mg biomass/l
PHYTO = phytoplankton in mg biomass/l

Subroutine RQUAL performs two tasks. First, RQUAL is responsible for calling the four subroutine groups which simulate the constituents listed above. These four groups and their functions are:

1. OXRX: simulate primary dissolved oxygen and biochemical oxygen demand balances
2. NUTRX: determine inorganic nitrogen and phosphorus balances
3. PLANK: simulate plankton populations and associated reactions
4. PHCARB: simulate pH and inorganic carbon species

The four groups are listed in their order of execution, and the execution of a group is dependent upon the execution of the groups listed above it. For example, subroutine group PHCARB cannot be activated unless OXRX, NUTRX, and PLANK are active. On the other hand, the reactions in OXRX can be performed without the reactions contained in the other three subroutine groups.

The other function of RQUAL is to determine the values for variables which are used jointly by the four subroutine groups. The following variables are evaluated:

1. AVVELE: the average velocity of water in the RCHRES in ft/s
2. AVDEPE: the average depth of water in the RCHRES in ft
3. DEPCOR: conversion factor from square meters to liters (used for changing areal quantities from the benthal surface to equivalent volumetric values based on the depth of water in the RCHRES)
4. SCRFAC: scouring factor to be used for calculation of benthal release rates of inorganic nitrogen, orthophosphorus, carbon dioxide, and biochemical oxygen demand

SCRFAC has one of two values depending on the average velocity of the water in the RCHRES. AVVELE is compared to the value of parameter SCRVEL, the user specified velocity at and above which scouring occurs. If AVVELE is less than the value of parameter SCRVEL, then SCRFAC is set equal to 1.0, and there is no increase of benthal release rates due to scouring. If AVVELE is greater than SCRVEL, SCRFAC is set equal to the value of parameter SCRMUL, which is a constant multiplication factor applied directly to the release rates to account for scouring by rapidly moving water.

4.2(3).7.1 Simulate Primary DO and BOD Balances (Subroutine Group OXRX of Module RCHRES)

Purpose

The purpose of this code is to simulate the primary processes which determine the dissolved oxygen concentration in a reach or mixed reservoir. Dissolved oxygen concentration is generally viewed as an indicator of the overall well-being of streams or lakes and their associated ecological systems. In relatively unpolluted waters, sources and sinks of oxygen are in approximate balance, and the concentration remains close to saturation.

By contrast, in a stream receiving untreated waste waters, the natural balance is upset, bacteria predominate, and a significant depression of dissolved oxygen results (O'Connor 1970).

Schematic View of Fluxes and Storages

Figures 4.2(3).7.1-1 and 4.2(3).7.1-2 illustrate the fluxes and storages modeled in this subroutine group. In order to account for temporal variations in oxygen balance, state variables for both dissolved oxygen and biochemical oxygen demand must be maintained. The state variable DOX represents the oxygen dissolved in water and immediately available to satisfy the oxygen requirements of the system. The BOD state variable represents the total quantity of oxygen required to satisfy the first-stage (carbonaceous) biochemical oxygen demand of dead nonrefractory organic materials entrained in the water.

Subroutine OXRX considers the following processes in determining oxygen balance:

1. longitudinal advection of DOX and BOD
2. sinking of BOD material
3. benthal oxygen demand
4. benthal release of BOD material
5. reaeration
6. oxygen depletion due to decay of BOD materials

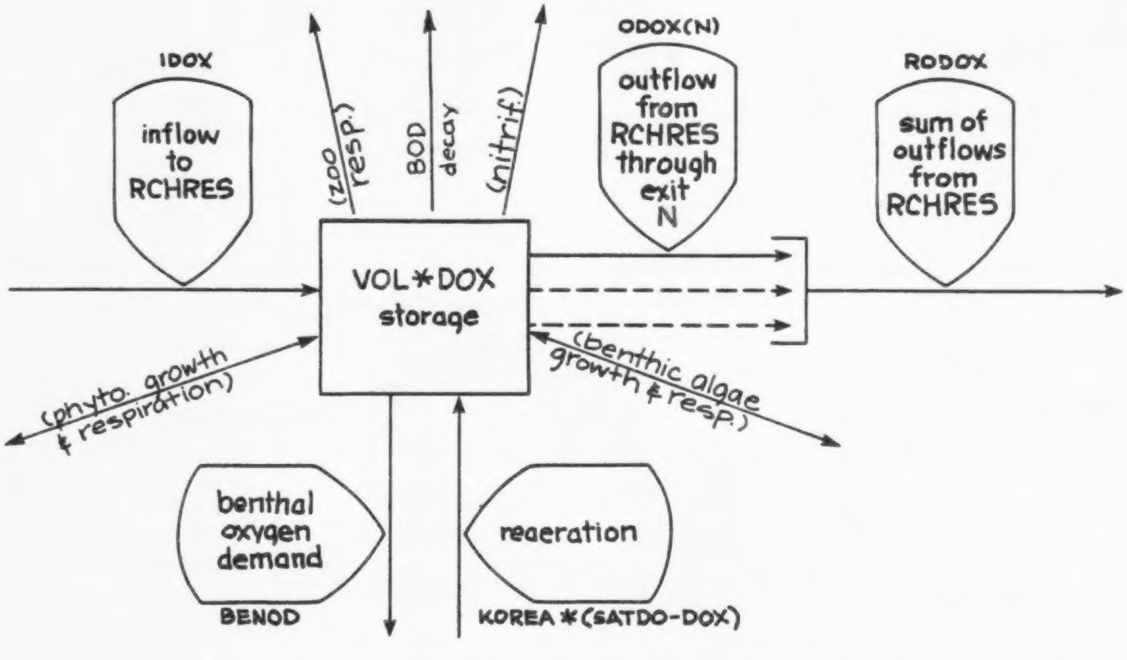


Figure 4.2(3).7.1-1 Flow diagram for dissolved oxygen in the OXRX subroutine group of the RCHRES Application Module.

Additional sources and sinks of DOX and BOD are simulated in other sections of the RCHRES module. If module section NUTRX (Section 4.2(3).7.2) is active, the effects of nitrification on dissolved oxygen and denitrification on BOD balance can be considered. If module section PLANK (Section 4.2(3).7.3) is active, the dissolved oxygen balance can be adjusted to account for photosynthetic and respiratory activity by phytoplankton and/or benthic algae and respiration by zooplankton. Adjustments to the BOD state variable in section PLANK include increments due to death of plankton and nonrefractory organic excretion by zooplankton.

Subroutine OXRX uses five subroutines to simulate dissolved oxygen and biochemical oxygen demand. Advection of DOX and BOD is performed by ADVECT (subroutine 4.2(3).3.1). Sinking of BOD material is carried out by SINK (subroutine 4.2(3).5.1.1). OXBEN calculates benthal oxygen demand and benthal release of BOD materials. Reaeration processes are simulated by utilizing OXREA, and BOD decay calculations are performed in BODDEC.

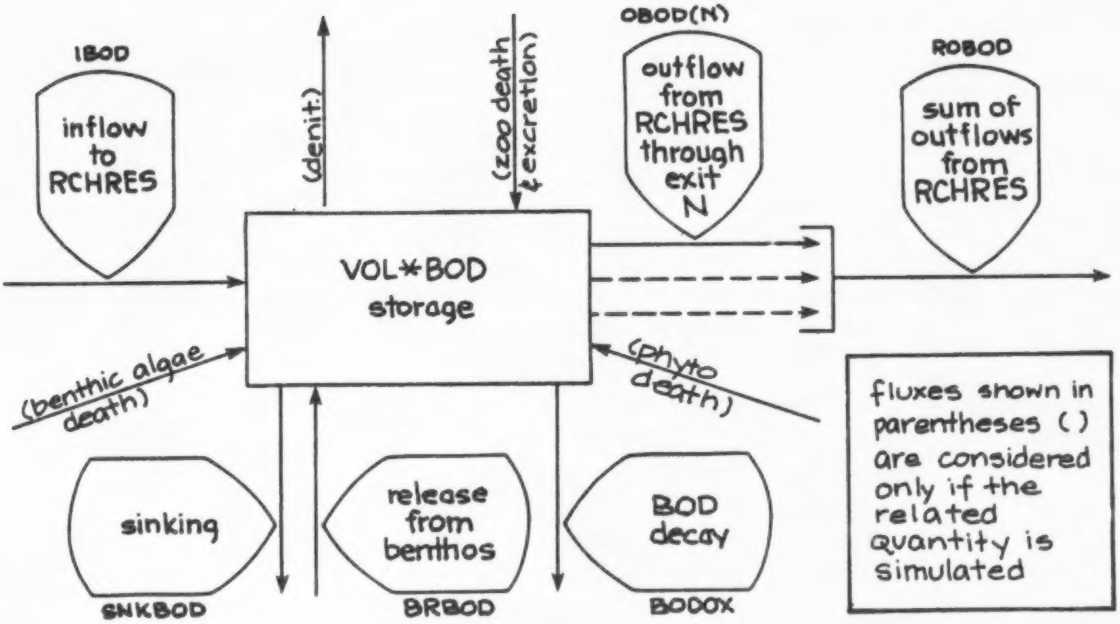


Figure 4.2(3).7.1-2 Flow diagram for biochemical oxygen demand in the OXRX subroutine group of the RCHRES Application module

4.2(3).7.1.1 Simulate Benthal Oxygen Demand and Benthal Release of BOD (subroutine OXBEN)

Purpose

OXBEN accounts for two possible demands exerted on available oxygen by the benthos. These two demands are categorized as benthal oxygen demand and benthal release of BOD materials. Benthal oxygen demand results from materials in the bottom muds which require oxygen for stabilization. This process results in a direct loss of oxygen from the RCHRES. The second demand on oxygen caused by the release and suspension of BOD materials is a less direct form of oxygen demand. This process increases the pool of BOD present in the RCHRES and exerts a demand on the dissolved oxygen concentration at a rate determined by the BOD decomposition kinetics.

Benthal Oxygen Demand

The user approximates the oxygen demand of the bottom muds at 20 degrees Celsius by assigning a value to BENOD for each RCHRES. The effects of temperature and dissolved oxygen concentration on realized benthal demand are determined by the following equation:

$$BENOX = BENOD*.05*TW*(1.0 - Exp(-1.22*DOX)) \qquad (1)$$

where:

BENOX = amount of oxygen demand exerted by the benthal muds in mg/m2 per interval
BENOD = reach dependent benthal oxygen demand at 20 degrees C in mg/m2 per interval
TW = water temperature in degrees C
DOX = dissolved oxygen concentration in mg/l

The first portion of the above equation (BENOD*.05*TW) proportionally adjusts the demand at 20 degrees Celsius to a demand at any temperature; that is, if water temperature is 10 degrees Celsius, the demand is half the value at 20 degrees. The second portion of the equation indicates that low concentrations of dissolved oxygen suppress realized oxygen demand. For example, 91 percent of BENOD may be realized at a dissolved oxygen concentration of 2 mg/l, 70 percent at 1 mg/l, and none if the waters are anoxic.

After the value of BENOX has been calculated, the dissolved oxygen state variable is updated:

$$DOX = DOX - BENOX*DEPCOR \qquad (2)$$

where:

DEPCOR = factor which converts from mg/m2 to mg/l, based on the average depth of water in the RCHRES during the simulation interval (DEPCOR is calculated in subroutine RQUAL 4.2(3).7)

Benthal Release of BOD

Bottom releases of BOD are a function of scouring potential and dissolved oxygen concentration. The equation used to calculate BOD release is:

```
     RELBOD = (BRBOD(1) + BRBOD(2)*Exp(-2.82*DOX))*SCRFAC                   (3)
```

where:

RELBOD = BOD released by bottom muds in mg/m2 per interval
BRBOD(1) = base release rate of BOD materials (aerobic conditions) in mg/m2 per interval
BRBOD(2) = increment to bottom release rate due to decreasing dissolved oxygen concentration
DOX = dissolved oxygen concentration in mg/l
SCRFAC = scouring factor dependent on average velocity of water (SCRFAC is calculated in subroutine RQUAL 4.2(3).7)

The above equation accounts for the fact that benthal releases are minimal during conditions of low velocity and ample dissolved oxygen. Under these conditions a thin layer of hardened, oxidized material typically retards further release of materials from the benthos. However, anaerobic conditions or increased velocity of overlying water disrupts this layer and release rates of BOD and other materials are increased. Solution of Equation 3 indicates that 6 percent of the incremental release rate (BRBOD(2)) occurs when 1 mg/l of dissolved oxygen is present, 75 percent occurs when 0.1 mg/l is present, and the entire increment occurs under anoxic conditions.

4.2(3).7.1.2 Calculate Oxygen Reaeration (subroutine OXREA)

Purpose

Various methods have been used to calculate atmospheric reaeration coefficients, and experience has shown that the most effective method of calculation in any given situation depends upon the prevalent hydraulic characteristics of the system (Covar 1976). Based upon user instructions, subroutine OXREA calculates oxygen reaeration by using one of four built-in solution techniques.

Approach

The general equation for reaeration is:

```
     DOX = DOXS + KOREA*(SATDO - DOXS)                                      (4)
```

where:

DOX = dissolved oxygen concentration after reaeration in mg/l
KOREA = reaeration coefficient (greater than zero and less than one)

SATDO = oxygen saturation level for given water temperature in mg/l
DOXS = dissolved oxygen concentration at start of interval in mg/l

Lake Reaeration

In a lake or reservoir, calculation of reaeration is dependent upon surface area, volume, and windspeed. The windspeed factor is determined using the following empirical relationship:

WINDF = WINDSP*(-.46 + .136*WINDSP) (5)

where:
WINDF = windspeed factor in lake reaeration calculation
WINDSP = windspeed expressed in m/sec

For low windspeeds, less than 6.0 m/s, WINDF is set to 2.0. The reaeration coefficient for lakes is calculated as:

KOREA = (.032808*WINDF*CFOREA/AVDEPE)*DELT60 (6)

where:
CFOREA = correction factor to reaeration coefficient for lakes; for lakes with poor circulation characteristics, CFOREA may be less than 1.0, and lakes with exceptional circulation characteristics may justify a value greater than 1.0 for CFOREA
AVDEPE = average depth of water in RCHRES during interval in ft
DELT60 = conversion from hourly time interval to simulation interval

Stream Reaeration

One of three approaches to calculating stream reaeration may be used:

1. Energy dissipation method (Tsivoglou-Wallace 1972). Oxygen reaeration is calculated based upon energy dissipation principles:

   KOREA = REAKT*(DELTHE/FLOTIM)*(TCGINV**(TW - 20.))*DELTS (7)

   where:
   REAKT = escape coefficient with a typical value between .054/ft and .110/ft.
   DELTHE = drop in energy line along length of RCHRES in ft
   FLOTIM = time of flow through RCHRES in seconds
   TCGINV = temperature correction coefficient for gas invasion rate with a default value of 1.047
   DELTS = conversion factor from units of per second to units of per interval

   DELTHE, the drop in elevation over the length of the RCHRES, is supplied by the user. REAKT, the escape coefficient, referred to in

Tsivoglou's work, is also supplied by the user. The value for FLOTIM is calculated by dividing the length of the RCHRES by the average velocity for the simulation interval. Tsivoglou's method of calculation is activated by setting the reaeration method flag (REAMFG) equal to 1.

2. Covar's method of determining reaeration (Covar 1976). Reaeration is calculated as a power function of hydraulic depth and velocity. The generalized equation used is:

```
KOREA = REAK*(AVVELE**EXPREV)*(AVDEPE**EXPRED)                (8)
        *(TCGINV**(TW - 20.))*DELT60
```

where:
KOREA = reaeration coefficient in units of per interval
REAK = empirical constant for reaeration equation, expressed in units of per hour
AVVELE = average velocity of water in ft/s
EXPREV = exponent to velocity function
AVDEPE = average water depth in ft
EXPRED = exponent to depth function
TCGINV = temperature correction coefficient for reaeration defaulted to 1.047
DELT60 = conversion factor from units of per hour to units of per interval

Depending on current depth and velocity, one of three sets of values for REAK, EXPREV, and EXPRED is used. Each set corresponds to an empirical formula which has proven accurate for a particular set of hydraulic conditions. The three formulas and their associated hydraulic conditions and coefficients are:

1. Owen's formula (1964). This formula is used for depths of less than 2 ft. For this formula, REAK = .906, EXPREV = 0.67, and EXPRED = -1.85.

2. Churchill's formula (1962). This formula is used for high velocity situations in depths of greater than 2 ft. For this formula, REAK = .484, EXPREV = .969, and EXPRED = -1.673.

3. O'Connor-Dobbins formula (1958). This formula is used for lower velocity situations in depths of greater than 2 ft. The coefficient values are: REAK = .538, EXPREV = 0.5, and EXPRED = -1.5.

This method of calculation of reaeration is activated by setting the reaeration method flag (REAMFG) equal to 2.

3. The user may select his own power function of hydraulic depth and velocity for use under all conditions of depth and velocity. In this case, he supplies values for REAK, EXPREV, and EXPRED. This option is selected by setting the reaeration method flag (REAMFG) to 3.

Reaeration may be modeled as a constant process for any given temperature. In this case, the user must supply a value for REAK, and a value of zero for both EXPREV and EXPRED. Note that subroutine OXREA requires input values for REAK, EXPREV, and EXPRED only if REAMFG is 3.

Calculation of Oxygen Saturation Concentration

The saturation concentration of dissolved oxygen is computed at prevalent atmospheric conditions by the equation:

SATDO = (14.652 + TW*(-.41022 + TW*(.007991 - .7777E-4*TW)))*CFPRES (9)

where:

SATDO = saturated conc of dissolved oxygen in mg/l
TW = water temperature in degrees C
CFPRES = ratio of site pressure to sea level pressure (CFPRES is calculated by the Run Interpreter dependent upon mean elevation of RCHRES)

4.2(3).7.1.3 Calculate BOD Decay (subroutine BODDEC)

Purpose

Subroutine BODDEC adjusts the dissolved oxygen concentration of the water to account for the oxygen consumed by microorganisms as they break down complex materials to simpler and more stable products. Only carbonaceous BOD is considered in this subroutine. The BOD decay process is assumed to follow first order kinetics and is represented by:

BODOX = (KBOD20*(TCBOD**(TW - 20.))*BOD (10)

where:

BODOX = quantity of oxygen required to satisfy BOD decay in mg/l per interval
KBOD20 = BOD decay rate at 20 degrees C
TCBOD = temperature correction coefficient, defaulted to 1.075
TW = water temperature in degrees C
BOD = BOD concentration expressed in mg/l

If there is not sufficient dissolved oxygen available to satisfy the entire demand exerted by BOD decay, the fraction which can be satisfied is subtracted from the BOD state variable, and the DOX variable is set to zero. The quantity of unsatisfied BOD decay for the interval is retained. If the algorithms which simulate denitrification in subroutine group NUTRX are active, oxygen produced through the denitrification process may be used to satisfy the remainder of the oxygen deficit caused by BOD decay. Any BOD which is still unsatisfied remains part of the BOD available for decay in future time steps.

4.2(3).7.2 Simulate Primary Inorganic Nitrogen and Phosphorus Balances (Subroutine Group NUTRX of Module RCHRES)

Purpose

This code simulates the primary processes which determine the balance of inorganic nitrogen and phosphorus in natural waters. When modeling the water quality of an aquatic system, consideration of both nitrogen and phosphorus is essential. Nitrogen, in its various forms, can deplete dissolved oxygen levels in receiving waters, stimulate aquatic growth, exhibit toxicity toward aquatic life, or present a public health hazard (EPA 1975). Phosphorus is vital in the operation of energy transfer systems in biota, and in many cases is the growth limiting factor for algal communities. Consequently, it is necessary to model phosphorus in any study concerned with eutrophication processes.

Schematic View of Fluxes and Storages

Figures 4.2(3).7.2-1 and 4.2(3).7.2-2 illustrate the fluxes and storages of four constituents which are introduced into the RCHRES modeling system in subroutine group NUTRX. In addition to these constituents, the state variables for dissolved oxygen and BOD are also updated. If subroutine group NUTRX is active (NUTFG = 1), nitrate will automatically be simulated; the user must specify whether or not nitrite, ammonia, and/or orthophosphorus are to be simulated in addition to nitrate by assigning appropriate values to NO2FG, NH3FG, and PO4FG in the User's Control Input. If all four constituents are simulated, subroutine NUTRX considers the following processes:

1. longitudinal advection of NO3, NO2, NH3, and PO4
2. benthal release of inorganic nitrogen and PO4 (if BENRFG = 1)
3. ammonia vaporization (if AMVFG = 1)
4. nitrification
5. denitrification (if DENFG = 1)
6. ammonification due to degradation of BOD materials

Additional sources and sinks of NO3, NH3, and PO4 are simulated in the PLANK section (4.2(3).7.3) of this module. If section PLANK is active, the state variables for these three constituents can be adjusted to account for nutrient uptake by phytoplankton and/or benthic algae, and for respiration and inorganic excretion by zooplankton.

Subroutine NUTRX utilizes five subroutines to simulate inorganic nitrogen and phosphorus. Advection of NO3, NO2, NH3, and PO4 is performed by ADVECT (4.2(3).3.1). BENTH determines the amount of inorganic nitrogen and phosphorus which is released to the overlying waters from the benthos. The nitrification and denitrification processes are simulated by NITRIF and DENIT, respectively. Finally, the production of inorganic nitrogen and phosphorus resulting from degradation of BOD materials is simulated by DECBAL.

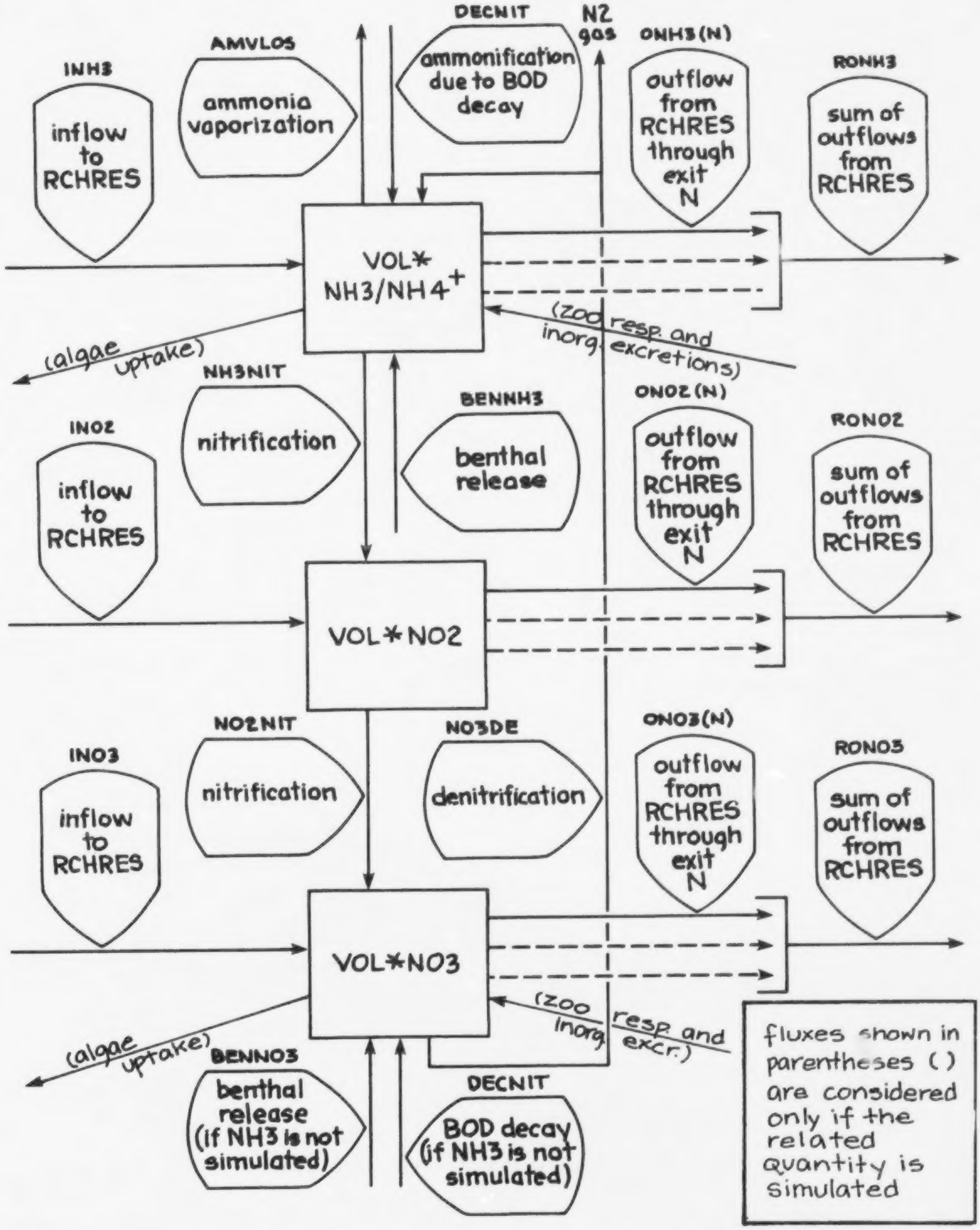

DECNIT
ammonification due to BOD decay
N2 gas
AMVLOS
ammonia vaporization
ONH3(N)
outflow from RCHRES through exit N
INH3
inflow to RCHRES
RONH3
sum of outflows from RCHRES
VOL* NH3/NH4+
(algae uptake)
(zoo resp. and inorg. excretions)
NH3NIT
nitrification
BENNH3
benthal release
ONO2(N)
outflow from RCHRES through exit N
INO2
inflow to RCHRES
RONO2
sum of outflows from RCHRES
VOL*NO2
NO2NIT
nitrification
NO3DE
denitrification
ONO3(N)
outflow from RCHRES through exit N
INO3
inflow to RCHRES
RONO3
sum of outflows from RCHRES
VOL*NO3
(algae uptake)
(zoo resp. and inorg. excr.)
BENNO3
benthal release (if NH3 is not simulated)
DECNIT
BOD decay (if NH3 is not simulated)
fluxes shown in parentheses ( ) are considered only if the related quantity is simulated

Figure 4.2(3).7.2-1 Flow diagram for inorganic nitrogen in the NUTRX subroutine group of the RCHRES Application Module

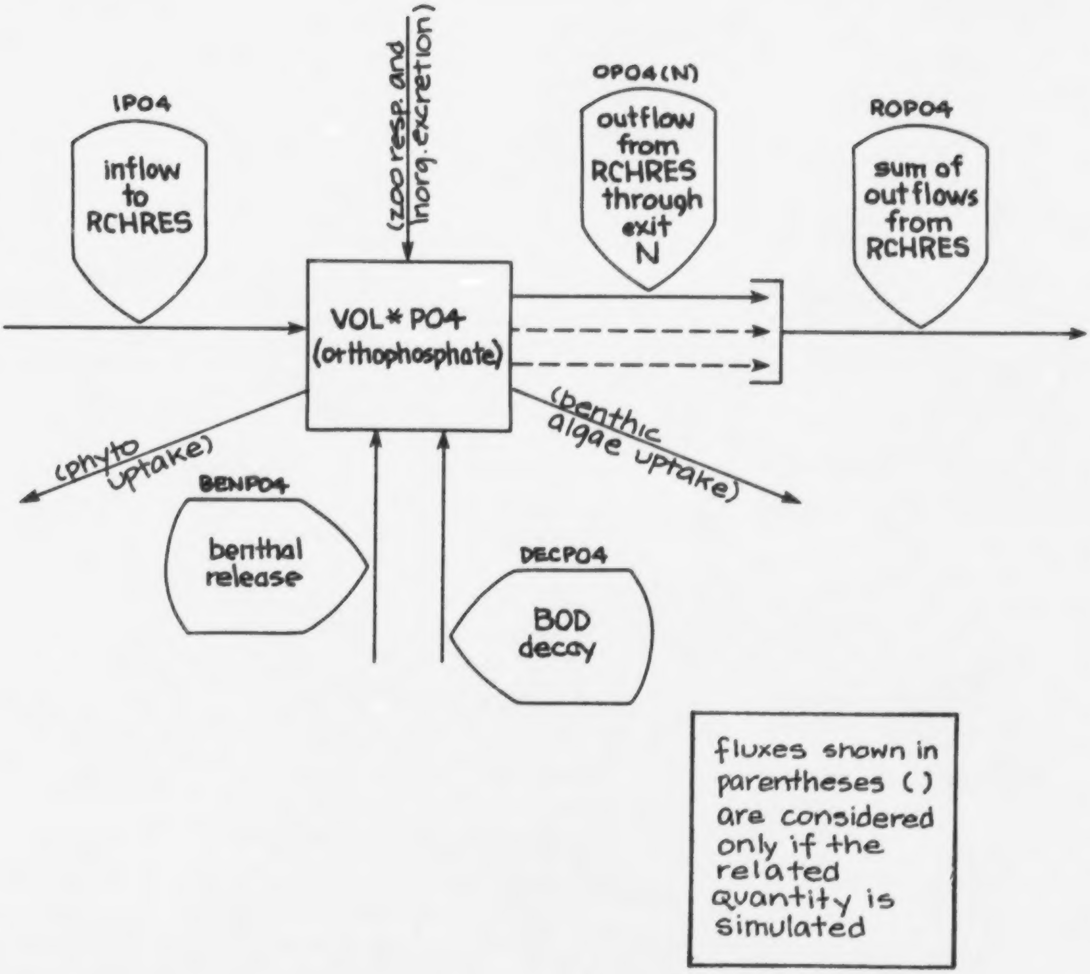

IPO4
inflow to RCHRES
(zoo resp. and inorg. excretion)
OPO4(N)
outflow from RCHRES through exit N
ROPO4
sum of outflows from RCHRES
VOL* PO4 (orthophosphate)
(phyto uptake)
(benthic algae uptake)
BENPO4
benthal release
DECPO4
BOD decay
fluxes shown in parentheses ( ) are considered only if the related quantity is simulated

Figure 4.2(3).7.2-2 Flow diagram for ortho-phosphate in the NUTRX group of the RCHRES Application Module.

Ammonia Vaporization

The amount of ammonia lost from the RCHRES due to ammonia vaporization is calculated by the following empirical relationship:

AMVLOS = 0.048*Exp(0.13*(TW - 20.))*NH3*DELT60 (1)

where:
AMVLOS = amount of ammonia vaporized expressed as mg NH3-N/interval
TW = water temperature in degrees C
NH3 = concentration of ammonia (mg/l)
DELT60 = conversion from units of per hour to units of per interval

Simulation of ammonia vaporization is activated by setting AMVFG equal to one in the User's Control Input.

4.2(3).7.2.1 Simulate Benthal Release of Constituents (subroutine BENTH)

Purpose

This subroutine checks to see whether present water conditions are aerobic or anaerobic, calculates benthal release for a constituent based on this check, and updates the concentration of the constituent.

Approach

The equation used to calculate release is:

RELEAS = BRCON(I)*SCRFAC*DEPCOR (2)

where:
RELEAS = amount of constituent released expressed in mg/l per interval
BRCON(I) = benthal release rate for constituent expressed as mg/m2 per interval
SCRFAC = scouring factor, dependent on average velocity of the water (SCRFAC is calculated in subroutine RQUAL (4.2(3).7))
DEPCOR = conversion factor from mg/m2 to mg/l (DEPCOR is calculated in subroutine RQUAL)

The dissolved oxygen concentration below which anaerobic conditions are considered to exist is determined by the input parameter ANAER. Two release rates are required for each of the constituents: one for aerobic conditions and one for anaerobic conditions. Typically, the aerobic release rate is less than the anaerobic rate, because a layer of oxidized materials forms on the benthal surface during aerobic periods, and this layer retards the release rate of additional benthal materials. BRCON(1) is the aerobic release rate and BRCON(2) is the anaerobic rate. The choice of which release rate is used, is determined by comparing the current value of DOX to ANAER.

If ammonia is simulated, the inorganic nitrogen release from the benthos is assumed to be in the form of ammonia, and the NH3 state variable is updated. If ammonia is not simulated, benthal release of inorganic nitrogen is assumed to be in the form of nitrate, and the NO3 state variable is updated. If orthophosphate is simulated, an additional call is made to BENTH to account for release of PO4.

Simulation of benthal release processes is activated by assigning a value of one to BENRFG in the User's Control Input.

4.2(3).7.2.2 Simulate Nitrification (subroutine NITRIF)

Purpose

NITRIF simulates the oxidation of ammonium and nitrite by chemoautotrophic bacteria. This oxidation provides energy for bacteria much the same way that sunlight provides energy for photosynthetic algae. The Nitrosomonas genera are responsible for conversion of ammonium to nitrite, and Nitrobacter perform oxidation of nitrite to nitrate. (It should be noted that no differentiation is made between ammonia and ammonium in HSPF.) Oxidation of inorganic nitrogen is dependent upon a suitable supply of dissolved oxygen; subroutine NITRIF does not simulate nitrification if the DO concentration is less than 2 mg/l.

Method

The rate of nitrification is represented by a first order equation in which nitrification is directly proportional to the quantity of reactant present, either ammonia or nitrite. The equation used to calculate the amount of NH3/NH4 oxidized to NO2 is:

```
NH3NIT = KNH320*(TCNIT**(TW - 20.))*NH3                      (3)
```

where:

NH3NIT = amount of NH3 oxidation expressed in mg NH3-N/l per interval

KNH320 = NH3 oxidation rate coefficient at 20 degrees C expressed in units of per interval

TCNIT = temperature correction coefficient, defaulted to 1.2

TW = water temperature in degrees C

NH3 = ammonia concentration in NH3-N/l

Similarly, if nitrite is simulated, the amount of nitrite oxidized to nitrate is determined by the equation:

```
NO2NIT = KNO220*(TCNIT**(TW - 20.))*NO2                      (4)
```

where:

NO2NIT = amount of NO2 oxidation expressed in mg NO2-N/l per interval

KNO220 = NO2 oxidation rate coefficient at 20 degrees C expressed in units of per interval
NO2 = nitrite concentration in mg NO2-N/l

The amount of oxygen used during nitrification is 3.43 mg oxygen per mg NH3-N oxidized to NO2-N, and 1.14 mg oxygen per mg NO2-N oxidized to NO3-N. In the RCHRES module, these figures are adjusted to 3.22 mg and 1.11 mg, respectively, to account for the effects of carbon dioxide fixation by bacteria (Wezerak and Gannon 1968). Thus, the oxygen demand due to nitrification is evaluated as:

$$DODEMD = 3.22*NH3NIT + 1.11*NO2NIT \tag{5}$$

where:
DODEMD = loss of dissolved oxygen from the RCHRES due to nitrification, expressed as mg O/l per interval

If the value of DODEMD is greater than available dissolved oxygen, the amounts of oxidation from NH3 to NO2 and from NO2 to NO3 are proportionally reduced, so that state variable DOX maintains a nonnegative value. If nitrite is not simulated, the calculated amount of oxidized ammonia is assumed to be fully oxidized to nitrate.

4.2(3).7.2.3 Simulate Denitrification (subroutine DENIT)

Purpose

DENIT simulates the reduction of nitrate by facultative anaerobic bacteria such as Pseudomonas, Micrococcus, and Bacillus. These bacteria can use NO3 for respiration in the same manner that oxygen is used under aerobic conditions. Facultative organisms use oxygen until the environment becomes nearly or totally anaerobic, and then switch over to NO3 as their oxygen source. In most cases the end product of denitrification is nitrogen gas, but in special cases the end product may be ammonia. If denitrification is simulated (DENFG = 1), the user must specify the end product of denitrification by assigning a value of zero or one to DENRFG in the User's Control Input. A zero value indicates the end product is nitrogen gas, and a value of one indicates ammonia.

Approach

Denitrification does not occur in the RCHRES module unless the potential BOD decay calculated in subroutine group OXRX was not fully satisfied. In such cases, the amount of nitrate which must be reduced to satisfy the remaining oxygen deficit is calculated as:

$$PNO3DE = -DEFOX*.218818 \tag{6}$$

where:
PNO3DE = nitrate requirement expressed as mg NO3-N/l per interval

-DEFOX = amount of unsatisfied BOD decay for interval expressed as mg O/l
.218818 = stoichiometric equivalence factor between nitrate and oxygen (production of 1 mg of oxygen results from reduction of .218818 mg of nitrate-nitrogen)

The actual amount of denitrification for the interval is calculated by the following equation:

NO3DE = PNO3DE*DEBAC (7)

where:
DEBAC = unitless factor which represents the relative abundance of denitrifying bacteria

The factor DEBAC is a fraction between zero and one. The factor is diminished by .004 per hour to account for death of bacteria. During anaerobic periods DEBAC is increased by .019 per hour to represent growth of bacteria. The model assumes no growth of denitrifying bacteria during aerobic periods.

If denitrification of 97 percent or less of the available nitrate can provide sufficient oxygen to compensate for the remaining oxygen deficit, then -DEFOX will be satisfied by denitrification.

4.2(3).7.2.4 Perform Materials Balance for Transformation from Organic to Inorganic Material (subroutine DECBAL)

Purpose

DECBAL adjusts the inorganic nitrogen and orthophosphorus state variables to account for decomposition of organic materials.

Method

In subroutine NUTRX the total BOD decay for the time interval is determined by summing the decay satisfied by dissolved oxygen and that satisfied by denitrification. At the same time the corresponding amounts of inorganic nitrogen and orthophosphorus produced by the decay are determined as:

DECNIT = BODOX*CVON (8)
DECPO4 = BODOX*CVOP (9)

where:
BODOX = total BOD decay expressed as mg O/l per interval
CVON = stoichiometric conversion factor from mg oxygen to mg nitrogen

CVOP = stoichiometric conversion factor from mg oxygen to mg phosphorus

The values for DECNIT and DECPO4 are passed to subroutine DECBAL. If ammonia is simulated, the value of DECNIT is added to the NH3 state variable; if not, DECNIT is added to the NO3 state variable. If orthophosphorus is simulated, the value of DECPO4 is added to the PO4 state variable.

4.2(3).7.3 Simulate Plankton Populations and Associated Reactions (Subroutine Group PLANK of Module RCHRES)

Purpose

PLANK simulates phytoplankton, zooplankton, and/or benthic algae.

Schematic View of Fluxes and Storages

Figures 4.2(3).7.3-1 through 4.2(3).7.3-4 illustrate the fluxes and storages of six constituents which are introduced into the RCHRES modeling system in subroutine PLANK. In addition to these constituents, the state variables for dissolved oxygen, biochemical oxygen demand, nitrate, ammonia, and orthophosphorus are also updated. If subroutine group PLANK is active (PLKFG = 1), dead refractory organics will automatically be simulated. The state variables for these organics are ORN (dead refractory organic nitrogen), ORP (dead refractory organic phosphorus), and ORC (dead refractory organic carbon). The user must specify whether or not phytoplankton, zooplankton, and/or benthic algae are simulated by assigning appropriate values to PHYFG, ZOOFG, and BALFG in the User's Control Input. The state variable PHYTO represents the free floating photosynthetic algae, ZOO represents the zooplankton which feed on PHYTO, and BENAL is the state variable for algae attached to the benthal surface.

Subroutine group PLANK is the largest and most complex of the code segments in the RCHRES module. PLANK uses twelve subroutines to perform simulation of the three types of plankton. Longitudinal advection of PHYTO and ZOO is performed by ADVPLK, a special advection routine for plankton. ORN, ORP, and ORC are advected by ADVECT. The sinking of PHYTO, ORN, ORP, and ORC is performed by SINK. The user controls the sinking rate of these constituents by assigning values to parameters PHYSET and REFSET in the User's Control Input. PHYSET is the rate of phytoplankton settling, and REFSET is the settling rate for all three of the dead refractory organic constituents. Advection and sinking are performed every interval of the simulation period. The remainder of the processes modeled in PLANK are only performed when the average depth of water in the RCHRES is at least 2 in. Experience has shown that the algorithms used to represent these processes are not accurate for excessively shallow waters. If 2 in. or more of water is present in the RCHRES, PLANK performs a series of operations which are necessary to determine the availability of light to support algal growth. First the light intensity at the RCHRES surface is calculated by the following equation:

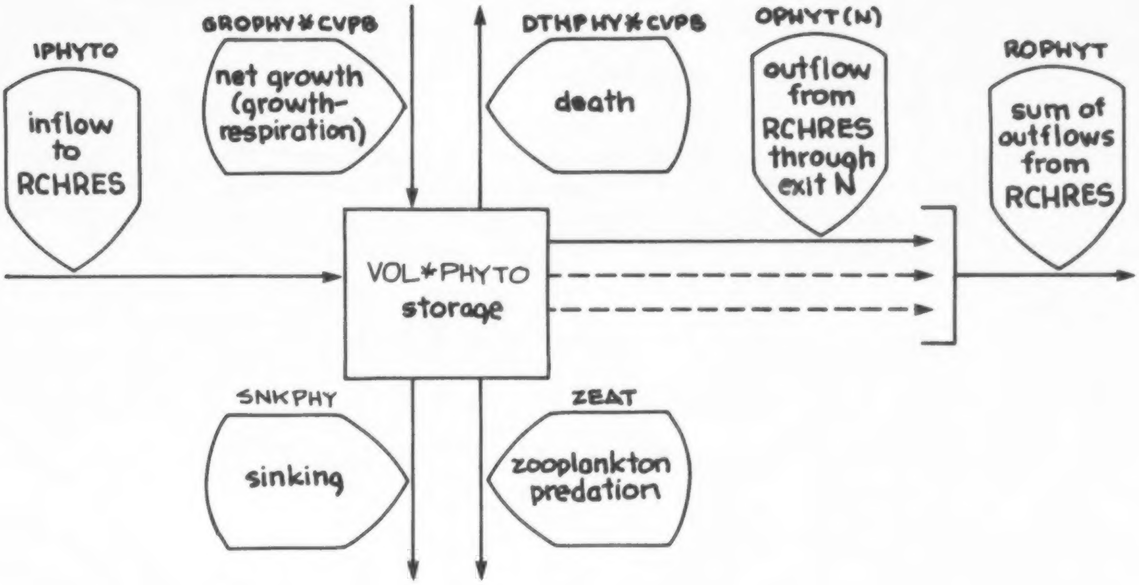


Figure 4.2(3).7.3-1 Flow diagram for phytoplankton in the PLANK section of the RCHRES Application Module.

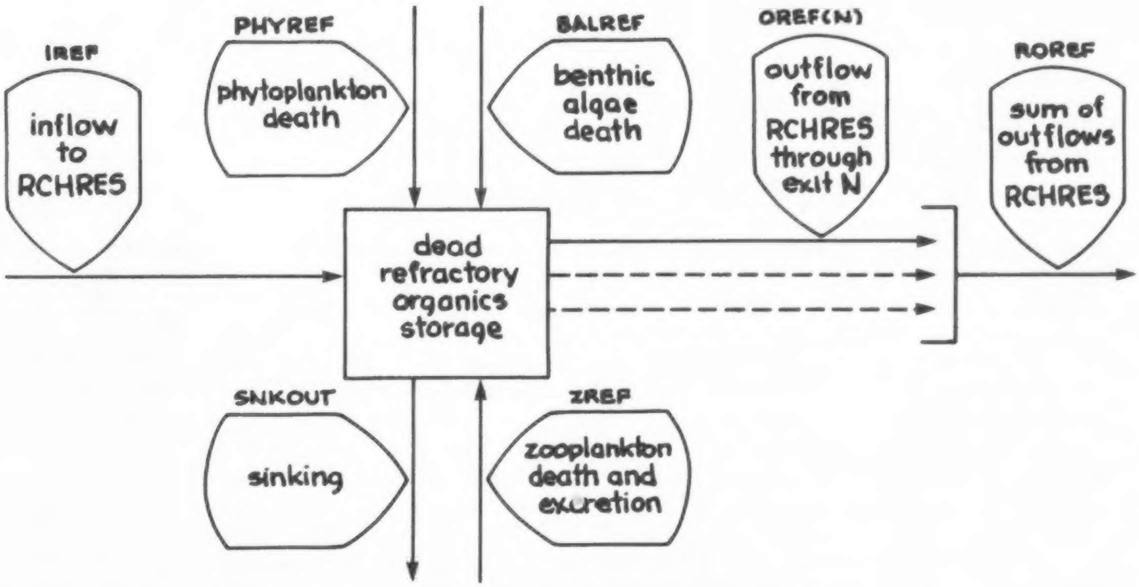

IREF
inflow to RCHRES
PHYREF
phytoplankton death
BALREF
benthic algae death
OREF(N)
outflow from RCHRES through exit N
ROREF
sum of outflows from RCHRES
dead refractory organics storage
SNKOUT
sinking
ZREF
zooplankton death and excretion

Figure 4.2(3).7.3-2 Flow diagram for dead refractory organics in the PLANK section of the RCHRES Application Module. This diagram illustrates the flow of ORN, ORP, and ORC through the RCHRES.

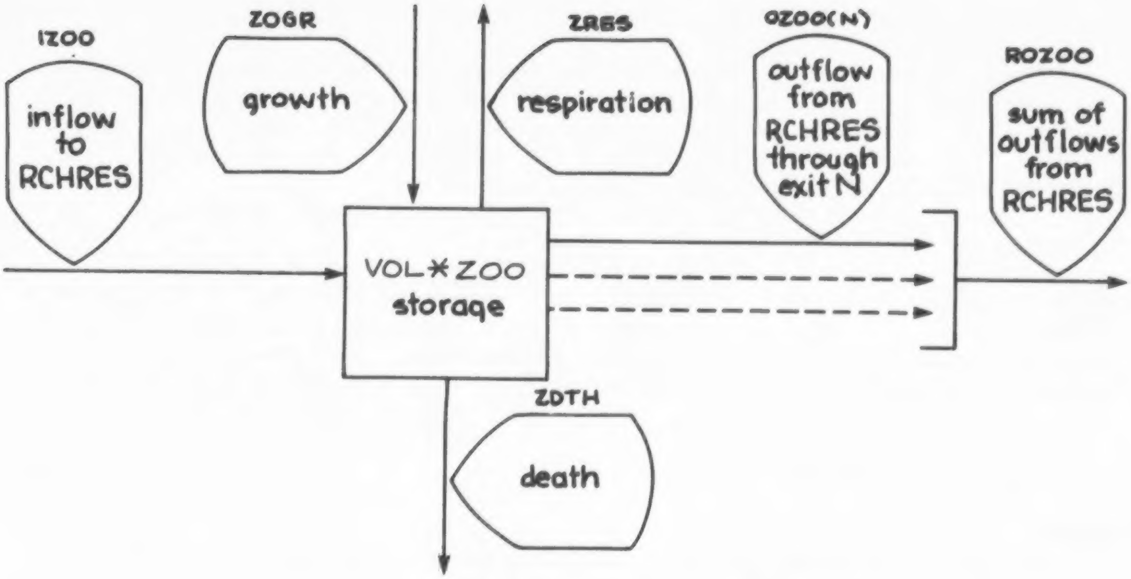


Figure 4.2(3).7.3-3 Flow diagram for zooplankton in the PLANK section of the RCHRES Application Module

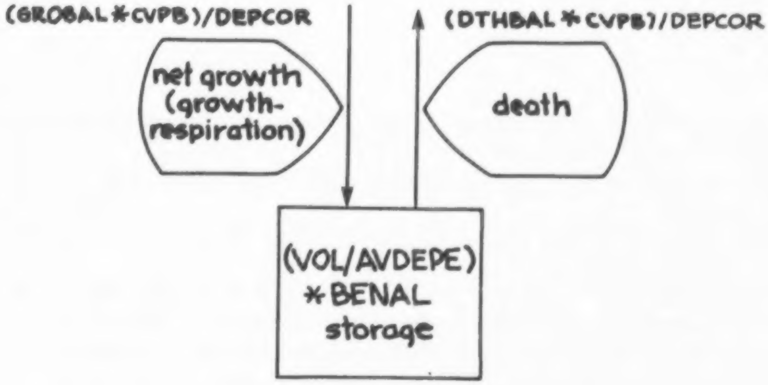


Figure 4.2(3).7.3-4 Flow diagram for benthic algae in the PLANK section of the RCHRES Application Module

INLIT = 0.97*CFSAEX*SOLRAD/DELT (1)

where:

INLIT = light intensity immediately below water surface, as langleys/min

0.97 = correction factor for surface reflection (assumed 3 percent)

CFSAEX = input parameter which specifies the ratio of radiation at water surface to gage radiation values. This factor also accounts for shading of the water body, eg. by trees

SOLRAD = solar radiation in langleys/interval

DELT = conversion from units of per interval to per minute

After light intensity at the water surface has been calculated, PLANK determines the factors which diminish the intensity of light as it passes downward from the surface. In addition to the natural extinction due to passage through water, extinction may result from interference caused by suspended sediment or phytoplankton. If SDLTFG is assigned a value of one, the contribution of washload to light extinction is calculated as:

EXTWSH = LITWSH*WASH (2)

where:

EXTWSH = increment to base extinction coefficient due to washload in units of per foot

LITWSH = multiplication factor to washload concentration (supplied in User's Control Input)

WASH = washload in mg/l

The contribution of suspended phytoplankton to light extinction is determined by the empirical relationship:

EXTCLA = .00452*PHYCLA (3)

where:

EXTCLA = increment to base extinction coefficient due to phytoplankton, in units of per foot

.00452 = multiplication factor to phytoplankton chlorophyll a concentration

PHYCLA = phytoplankton concentration as micromoles chlorophyll a/l

After values for INLIT, EXTWSH, and EXTCLA have been calculated, PLANK calls subroutine LITRCH to determine the light correction factor to algal growth and the amount of light available to phytoplankton and benthic algae. Once these calculations have been completed, PLANK checks a series of flags to determine which types of plankton are to be simulated. If PHYFG is assigned a value of one, simulation of phytoplankton is performed by a group of six subroutines. Zooplankton are simulated by a group of three subroutines if ZOOFG is given a value of one. Zooplankton simulation can be performed only if the phytoplankton section is active. Finally, a value of one for BALFG activates benthic algae simulation by a group of five subroutines. The organization of the subroutines in the PLANK group is clarified by referring to structure charts 4.2(3).7.3 through 4.2(3).7.3.5 in Part D of this document.

4.2(3).7.3.1 Advect Plankton (subroutine ADVPLK)

Purpose

ADVPLK performs the advection of phytoplankton and zooplankton. The normal advection method (subroutine ADVECT) used in the RCHRES module assumes that each constituent concentration is uniform throughout the RCHRES. This assumption is not valid for plankton. Both phytoplankton and zooplankton locate their breeding grounds near the channel boundaries. Since the water near the boundaries moves downstream much more slowly than the mean water velocity, the plankton populations have a much longer residence time in the RCHRES than would be indicated by the mean flow time. The geographical extent of the plankton breeding grounds is inversely related to the flow rate. At low flows, large areas of slow moving waters which are suitable for breeding exist along the channel boundaries. As flow rates increase, more and more of these areas are subject to flushing. The special advection routine is critical to plankton simulation, because the only source of plankton is within the reach network. Thus an upstream RCHRES with no plankton inflows can maintain a significant plankton population only if the growth rate of plankton exceeds the rate at which plankton are advected out of the RCHRES. Since biological growth rates are typically much slower than "normal" advection rates, few free-flowing RCHRES's could maintain a plankton population without the use of the special advection routine.

Method

Figure 4.2(3).7.3-5 illustrates the relationships used to perform plankton advection.

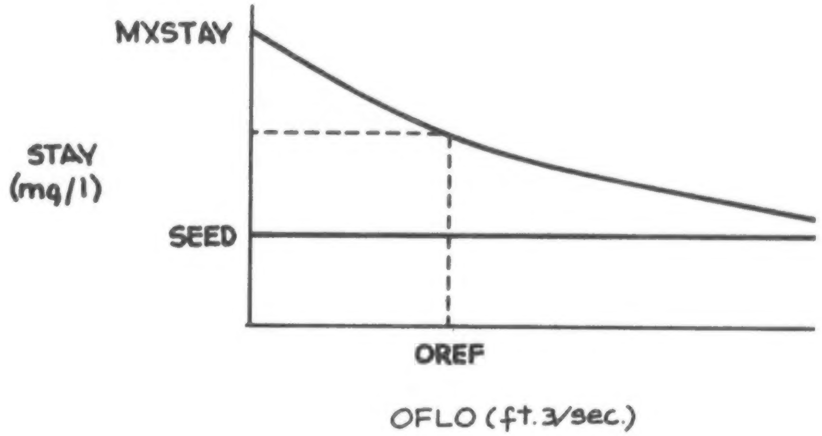


Figure 4.2(3).7.3-5 Relationship of parameters for special advection of plankton

ADVPLK assumes that a certain concentration of plankton (STAY) is not subject to advection, but any excess of organisms will be advected in the normal way. A small population (SEED) of plankton are never subject to advection, even during the periods of greatest flow. The maximum concentration of plankton which is not subject to advection (MXSTAY) occurs during low flow conditions. Each simulation interval ADVPLK calculates STAY based on the values of these two parameters and OREF. OREF is the outflow rate at which STAY has a value midway between SEED and MXSTAY. First, the average flow rate through the RCHRES for the interval is calculated:

$$\mathrm{OFLO} = (\mathrm{SROVOL} + \mathrm{EROVOL})/\mathrm{DELTS} \tag{4}$$

where:

OFLO = average flow rate (ft3/s or m3/s)
DELTS = number of seconds per interval
SROVOL and EROVOL are as defined in Section 4.2(3).2

The concentration of plankton which are not subject to advection is then determined:

$$\mathrm{STAY} = (\mathrm{MXSTAY} - \mathrm{SEED})*(2.0**(-\mathrm{OFLO}/\mathrm{OREF})) + \mathrm{SEED} \tag{5}$$

where:

STAY = plankton concentration not advected in mg/l
MXSTAY = maximum concentration not subject to advection
SEED = concentration of plankton never subject to advection
OREF = outflow rate at which STAY has a value midway between SEED and MXSTAY (ft3/s or m3/s)

The amount of plankton not subject to advection is converted to units of mass (MSTAY) by multipling STAY by the volume in the RCHRES at the start of the interval (VOLS). The concentration of plankton which are advected is:

$$\mathrm{PLNKAD} = \mathrm{PLANK} - \mathrm{STAY} \tag{6}$$

ADVPLK calls subroutine ADVECT (4.2(3).3.1) to perform longitudinal advection of the quantity PLNKAD. The updated value of PLNKAD is then added to the amount of plankton which did not undergo advection to determine the concentration of plankton in the RCHRES at the end of the interval:

$$\mathrm{PLANK} = \mathrm{PLNKAD} + \mathrm{MSTAY}/\mathrm{VOL} \tag{7}$$

where:

PLANK = concentration of plankton at end of interval
PLNKAD = concentration of advected plankton which remain in RCHRES
MSTAY = mass of plankton not advected
VOL = volume in RCHRES at end of interval

If the concentration of plankton in the RCHRES at the start of the interval is less than the value assigned to SEED, advection of plankton is not performed in the RCHRES, and the value of PLANK at the end of the interval is calculated as:

PLANK = (MSTAY + IPLANK)/VOL (8)

where:
IPLANK = mass of plankton which enters RCHRES during interval

4.2(3).7.3.2 Calculate Light-related Information Needed for Algal Simulation (subroutine LITRCH)

Purpose

Subroutine LITRCH determines the light correction factor to algal growth and the amount of light available to phytoplankton and benthic algae.

Method

The overall light extinction factor for the interval is obtained by adding EXTWSH and EXTCLA to the base extinction coefficient (EXTB). The value of EXTB is assumed constant for a particular RCHRES and must be assigned in the User's Control Input. The resulting sum (EXTCO) is used to calculate the euphotic depth, which is the distance below the surface of the water body at which 1 percent of the light incident on the surface is still available:

EUDEP = 4.60517/EXTCO (9)

where:
EUDEP = euphotic depth in ft
EXTCO = total light extinction coefficient in units of per foot

HSPF assumes that growth of algae occurs only in the euphotic zone (that is, the water above euphotic depth). When EUDEP has been calculated, it is possible to assign a value to CFLIT, the light correction factor to algal growth. A value of 1.0 is assigned to CFLIT if the calculated euphotic zone includes all the water of the RCHRES. CFLIT = EUDEP/AVDEPE, if the euphotic depth is less than the average depth of water (AVDEPE). CFLIT is used in subroutine ALGRO, to adjust the computed rate of algal growth.

Finally, the amount of light available to phytoplankton and benthic algae is calculated. The equation used to calculate the amount of light available to phytoplankton assumes that all phytoplankton are at mid-depth in the RCHRES:

PHYLIT = INLIT*Exp(-EXTCO*(.5*AVDEPE)) (10)

where:
PHYLIT = light available to phytoplankton in langleys/min
INLIT = light available at water surface in langleys/min
EXTCO = light extinction coefficient in /ft
AVDEPE = average depth of water in the RCHRES in ft

The equation used to calculate the amount of light available to benthic algae assumes that all benthic algae are at AVDEPE below the surface of the RCHRES:

$$BALLIT = INLIT*Exp(-EXTCO*AVDEPE) \qquad (11)$$

4.2(3).7.3.3 Simulate Phytoplankton (subroutine PHYRX)

Purpose

PHYRX simulates the algae which float in the waters of a RCHRES. Because these organisms use energy from light to produce organic matter, they are called primary producers and are considered the first trophic level in the aquatic ecosystem. The biological activity of the ecosystem is dependent upon the rate of primary production by these photosynthetic organisms. The activities of the phytoplankton are in turn affected by the physical environment. Through the process of photosynthesis, phytoplankton consume carbon dioxide and release oxygen back into the water. At the same time, algal respiration consumes oxygen and releases carbon dioxide. Phytoplankton reduce the concentration of nutrients in the water by consuming phosphates, nitrate, and ammonia. Through assimilation these nutrients are transformed into organic materials which serve as a food source for members of higher trophic levels. A portion of the organic matter which is not used for food decomposes, which again affects the oxygen and nutrient concentrations in the water. Where the phytoplankton population has grown excessively, much of the available oxygen supply of the water may be depleted by decomposition of dead algae and respiration. In this situation, phytoplankton place a serious stress upon the system.

Approach

To describe quantitatively the the dynamic behavior of phytoplankton populations, a number of assumptions must be made. PHYRX treats the entire phytoplankton population as if it were one species, and the mean behavior of the population is described through a series of generalized mathematical formulations. While such an approach obscures the behavior of individual species, the overall effect of the phytoplankton population on the quality of the water can be modeled with reasonable accuracy.

The HSPF system assumes that biomass of all types (phytoplankton, zooplankton, benthic algae, dead organic materials) has a consistent chemical composition. The user specifies the biomass composition by indicating the carbon:nitrogen:phosphorus ratio and the percent-by-weight carbon. This is done by assigning values to the following parameters:

1. CVBPC: number of moles of carbon per mole of phosphorus in biomass (default = 106)

2. CVBPN: number of moles of nitrogen per mole of phosphorus in biomass (default = 16)
3. BPCNTC: percentage of biomass weight which is carbon (default = 49)

The algorithms used in PHYRX and its subroutines require that the phytoplankton population be expressed in units of micromoles of phosphorus per liter. PHYRX converts the value for state variable PHYTO in milligrams biomass per liter into micromoles phosphorus per liter and assigns this value to the internal state variable STC (standing crop).

PHYRX uses five subroutines to simulate phytoplankton. ALGRO computes unit growth and respiration rates and determines the growth limiting factor for the phytoplankton. If the amount of growth exceeds the amount of respiration for the interval, GROCHK adjusts growth to account for nutrient limitations. PHYDTH calculates the amount of death occurring during the interval. State variables ORN, ORP, ORC, and BOD are updated by ORGBAL to account for materials resulting from phytoplankton death. Finally, NUTRUP adjusts the values for PO4, NO3, and NH3 to account for uptake of nutrients by phytoplankton. In addition to these updates, the dissolved oxygen state variable is adjusted in PHYRX to account for the net effect of phytoplankton photosynthesis and respiration:

$$DOX = DOX + (CVPB*CVBO*GROPHY) \qquad (12)$$

where:

CVPB = conversion factor from micromoles phosphorus to mg biomass
CVBO = conversion factor from mg biomass to mg oxygen
GROPHY = net growth of phytoplankton as micromoles phosphorus/l per interval

After all the operations in PHYRX and its subroutines have been performed, the value of STC is converted back into units of milligrams biomass per liter and becomes the updated value of PHYTO.

4.2(3).7.3.3.1 Calculate Unit Growth and Respiration Rates for Algae (subroutine ALGRO)

Purpose

ALGRO calculates the unit growth rate of algae based on light, temperature, and nutrients. Each time step ALGRO determines the rate limiting factor for growth and passes a label which identifies the limiting factor to the subroutines responsible for printed output. The labels and their meanings are as follows:

'LIT'	Growth is light limited.
'NON'	Insufficient nutrients are available to support growth.

'TEM'	Water temperature does not allow algal growth.
'NIT'	Growth is limited by availability of inorganic nitrogen.
'PO4'	Growth is limited by availability of orthophosphorus.
'NONE'	There is no limiting factor to cause less than maximal growth.
'WAT'	Insufficient water is available to support growth.

ALGRO is also responsible for calculating the unit respiration rate for algae. This subroutine is used in the simulation of both phytoplankton and benthic algae.

Approach

ALGRO performs a series of initial checks to determine whether or not conditions are suitable for growth during the interval. If the light intensity for the interval is less than .001 langleys/min, insufficient light is available for growth, and growth is not calculated. Likewise, if the concentration of either inorganic nitrogen or orthophoshorus is less than .001 mg/l, no growth occurs. If these checks indicate that conditions are suitable for growth, ALGRO next determines the effects of water temperature on the growth potential.

Temperature Control

The user specifies the temperature preferences of the algae by assigning values to three parameters: TALGRL, TALGRM, and TALGRH. If the water temperature is less than the value assigned to TALGRL or greater than the value assigned to TALGRH, no growth occurs. For water temperatures between TALGRL and TALGRH, a correction factor to maximum growth rate (MALGR) is calculated. This correction factor increases in value linearly from 0.0 at TALGRL to 1.0 at TALGRM. Thus, TALGRM specifies the minimum temperature at which growth can occur at a maximum rate. ALGRO assumes that there is no temperature retardation of maximum growth rate for temperatures between TALGRM and TALGRH. The temperature corrected maximum growth rate is:

$$MALGRT = MALGR*TCMALG \tag{13}$$

where:

MALGRT = temperature corrected maximum algal growth rate in units of per interval
MALGR = maximum unit growth rate for algae
TCMALG = temperature correction to growth (TCMALG has a value between 0.0 and 1.0)

Once the temperature correction to potential growth rate has been made, ALGRO uses Monod growth kinetics with respect to orthophosphorus, inorganic nitrogen, and light intensity to determine the actual growth rate. The procedure taken in ALGRO is to consider each possible limiting factor separately to determine which one causes the smallest algal growth rate during each simulation interval. This method does not preclude that interactions between factors affect the actual growth rate; in cases where it has been established that there is such an interaction, as in the uptake of phosphate, the phenomena are included in the model. If none of the factors considered is limiting, growth will be maximal and temperature dependent.

Phosphorus Limited Growth

Algae are dependent upon uptake of orthophosphorus to provide the continual supply of phosphorus necessary for ordinary cellular metabolism and reproductive processes. In phosphorus limited situations, the resultant growth rate has been shown to be dependent not only on the concentration of phosphate ions, but on nitrate concentration as well (DiToro, et al. 1970). The phosphorus limited growth rate is determined by:

$$\mathrm{GROP = MALGRT*PO4*NO3/((PO4 + CMMP)*(NO3 + CMMNP))} \qquad (14)$$

where:

GROP = unit growth rate based on phosphorus limitation expressed in units of per interval

MALGRT = temperature corrected maximum algal growth rate

PO4 = orthophosphorus concentration in mg P/l

NO3 = nitrate concentration in mg N/l

CMMP = orthophosphorus Michaelis-Menten constant for phosphorus limited growth in mg P/l (CMMP is defaulted to .015 mg P/l)

CMMNP = nitrate Michaelis-Menten constant for phosphorus limited growth in mg N/l (CMMNP is defaulted to .0284 mg N/l)

Nitrogen Limited Growth

Nitrogen is essential to algae for assimilation of proteins and enzymes. In the form of nitrate, nitrogen serves as the essential hydrogen acceptor in the metabolic pathways which enable organisms to grow. ALGRO allows for two different sources of inorganic nitrogen. If ammonia is being simulated and a value of one is assigned to the nitrogen source flag (NSFG), both ammonia and nitrate are used by the algae to satisfy their nitrogen requirements. Otherwise, only nitrate is considered in the kinetics formulations. High ratios of ammonia to nitrate have been found to retard algal growth. If a value of one is assigned to the ammonia retardation flag (AMRFG), this phenomenon is simulated according to the equation:

$$\mathrm{MALGN = MALGRT - 0.757*NH3 + 0.051*NO3} \qquad (15)$$

where:

MALGN = maximum unit growth rate corrected for ammonia retardation in units of per interval

MALGRT = temperature corrected maximum unit growth rate

Nitrogen limitation on growth is calculated by the equation:

$$\mathrm{GRON} = \mathrm{MALGN} * \mathrm{MMN}/(\mathrm{MMN} + \mathrm{CMMN}) \tag{16}$$

where:

GRON = unit growth rate based on nitrogen limitation in units of per interval

MALGN = maximum unit growth rate (MALGN has the same value as MALGRT if AMRFG is set to zero)

MMN = total pool of inorganic nitrogen considered available for growth

CMMN = Michaelis-Menten constant for nitrogen limited growth in mg N/l (CMMN is defaulted to .045 mg N/l)

Light Limited Growth

The equation used to determine the limitation on growth rate imposed by light intensity was derived by Dugdale and Macisaac (1971) based on uptake rates of inorganic nitrogen under varying light intensities:

$$\mathrm{GROL} = \mathrm{MALGRT} * \mathrm{LIGHT}/(\mathrm{CMMLT} + \mathrm{LIGHT}) \tag{17}$$

where:

GROL = unit growth rate based on light limitation in units of per interval

MALGRT = temperature corrected maximum unit growth rate in units of per interval

LIGHT = light intensity available to algae in RCHRES in langleys/min

CMMLT = Michaelis-Menten constant for light limited growth in langleys/min (CMMLT is defaulted to .033 langleys/min)

Algal Respiration

Algal respiration is dependent upon water temperature and is calculated by the equation:

$$\mathrm{RES} = \mathrm{ALR20} * (\mathrm{TW}/20.) \tag{18}$$

where:

RES = unit algal respiration rate in units of per interval

ALR20 = unit respiration rate at 20 degrees C

TW = water temperature in degrees C

4.2(3).7.3.3.2 Check Nutrients Required for Computed Growth (subroutine GROCHK)

GROCHK assures that a minimum concentration of .001 mg/l of each nutrient remains in the RCHRES waters after growth occurs. If this condition is not satisfied, the computed growth rate is adjusted accordingly. Orthophosphorus and inorganic nitrogen are always considered as nutrients. If pH is simulated (PHFG = 1), the user may specify that carbon dioxide concentration also be considered as a limiting nutrient by setting the value of DECFG equal to zero.

4.2(3).7.3.3.3 Calculate Phytoplankton Death (subroutine PHYDTH)

Purpose

PHYDTH calculates algal death each interval by using one of two unit death rates specified in the User's Control Input. ALDL, the low unit death rate, is used when environmental conditions encourage sustained life. In situations where nutrients are scarce or the phytoplankton population becomes excessive, ALDH, the high algal death rate, is used.

Method

The high algal death rate, which has a default value of .01/hr, is used if any one of three conditions exists:

1. the concentration of PO4 is less than the value of parameter PALDH
2. the concentration of inorganic nitrogen is less than the value of parameter NALDH
3. the concentration of phytoplankton is greater than the value of parameter CLALDH

Regardless of whether these tests indicate that ALDH or ALDL should be used, an additional increment to death occurs if anaerobic conditions prevail during the interval. The increment to death rate due to anaerobic conditions is determined by the value of parameter OXALD. The amount of phytoplankton death which occurs during the interval is calculated as:

$$DTHPHY = ALD*STC \qquad (19)$$

where:

DTHPHY = amount of phytoplankton death as micromoles P/l.interval
ALD = unit algal death rate determined by environmental conditions in units of per interval
STC = concentration of phytoplankton as micromoles P/l

4.2(3).7.3.3.4 Perform Materials Balance for Transformation from Living to Dead Organic Material (subroutine ORGBAL)

Purpose

ORGBAL increments the concentrations of dead organics to account for plankton death. Plankton death may either be algal death, zooplankton death, or phytoplankton ingested by zooplankton but not assimilated. In each case in which ORGBAL is called, the increments to ORP, ORN, ORC, and BOD are calculated in the subroutine which makes the call and passed on to ORGBAL. ORGBAL is merely a service program which performs the additions to these state variables.

4.2(3).7.3.3.5 Perform Materials Balance for Transformation from Inorganic to Organic Materials (subroutine NUTRUP)

Purpose

NUTRUP adjusts the concentrations of inorganic chemicals to account for net growth of algae. Net growth may be either positive or negative depending on the relative magnitude of growth and respiration. The state variables which are updated by NUTRUP include PO4, NO3, NH3, and CO2.

Method

The adjustments to PO4 and CO2 are straightforward. The PO4 state variable is always updated; the CO2 state variable is only updated if pH is simulated (PHFG = 1) and carbon dioxide is considered as a limiting nutrient (DECFG = 0). Adjustment of the inorganic nitrogen state variables is more complex. If ammonia is not specified as a source of inorganic nitrogen for growth (NSFG = 0), only the NO3 state variable is updated to account for net growth. If ammonia is considered a nutrient (NSFG = 1), negative net growth is accounted for by adding the total flux of nitrogen to the NH3 state variable. If net growth is positive, a portion of the nitrogen flux is subtracted from both the NO3 and NH3 state variables. The relative proportions of NO3 and NH3 are governed by the value of parameter ALNPR, which is the fraction of nitrogen requirements for growth which are preferably satisfied by nitrate.

4.2(3).7.3.4 Simulate Zooplankton (subroutine ZORX)

Purpose

ZORX simulates the growth and death of zooplankton, and the resultant changes in the biochemical balance of the RCHRES. Zooplankton play an important role in determining the water quality of rivers and lakes. By feeding on the algal, bacterial, and detrital mass, they are a natural regulator in the aquatic environment. At the same time zooplankton are a source of food material for higher trophic levels such as fish. Through excretion, zooplankton provide nutrients for phytoplankton growth. HSPF is only concerned with those zooplankton which feed on phytoplankton, although in reality zooplankton may be herbivores, omnivores, or carnivores.

Schematic View of Fluxes and Storages

Figure 4.2(3).7.3-3 illustrates the fluxes and storage of zooplankton modeled in ZORX. In addition to zooplankton, the state variables for dissolved oxygen, biochemical oxygen demand, ammonia, nitrate, orthophosphate, and refractory organics are also updated. Subroutine ZORX considers the following processes:

1. filtering and ingestion of phytoplankton by zooplankton
2. assimilation of ingested materials to form new zooplankton biomass
3. zooplankton respiration
4. inorganic and organic zooplankton excretion
5. zooplankton death

Filtering and Ingestion

The amount of phytoplankton ingested per milligram zooplankton is calculated by the equation:

```
ZOEAT = ZFIL20*(TCZFIL**(TW - 20.))*PHYTO                        (20)
```

where:

ZOEAT = unit ingestion rate in mg phyto/mg zoo per interval
ZFIL20 = zooplankton filtering rate at 20 degrees C as liters filtered/mg zoo per interval
TCZFIL = temperature correction coefficient for filtering
TW = water temperature in degrees C
PHYTO = phytoplankton concentration in mg phyto/l

The filtering rate is dependent upon water temperature and phytoplankton concentration. Rates for most biological activities double for every 10 degrees Celsius increase in temperature. The filtering rate meets this criterion if the default value of 1.17 is used for the temperature correction coefficient TCZFIL.

When the phytoplankton biomass is below a critical concentration, the unit filtering rate will be maximal and constant. As the phytoplankton biomass increases above the critical concentration, the limiting rate is dependent on ingestive and digestive capabilities, and not on the concentration of the food source. Under these conditions, the filtering rate decreases proportionally such that the algal biomass ingested remains constant at the value of the parameter MZOEAT, which is defaulted to 0.055 mg phytoplankton/mg zooplankton/hr. The code simulates this by reducing ZOEAT to MZOEAT, if equation 20 gives a value greater than MZOEAT.

HSPF assumes that the filtering activities of zooplankton are 100 percent efficient; that is, the zooplankton ingest all of the food which is contained in the water which they filter. The total amount of phytoplankton ingested by the zooplankton is calculated as:

$$ZEAT = ZOEAT*ZOO \quad (21)$$

where:

ZEAT = ingested phytoplankton in mg biomass/l per interval
ZOEAT = unit ingestion rate
ZOO = zooplankton concentration in mg biomass/l

ZORX checks that the calculated amount of ingestion does not reduce the phytoplankton population to less than 0.0025 micromoles of phosphorus per liter; if it does, the ingestion rate is adjusted to maintain a phytoplankton concentration at this level.

Assimilation

Assimilation is the process by which ingested phytoplankton are converted to new zooplankton mass. The process of assimilation is never 100 percent efficient in biological systems. Unassimilated food is excreted as organic and inorganic waste products. Zooplankton assimilation efficiency is dependent upon quality and concentration of food. High quality food is assimilated at high efficiency, whereas low quality food is mostly excreted as waste resulting in low assimilation efficiency. The relationship between food concentration and assimilation efficiency is more complex. If the concentration of available food and the filtering rate of an organism are such that the organism ingests more food than can be readily used for growth and metabolism, the organism's assimilation efficiency decreases. The model represents the effect of food quality and concentration on assimilation as shown in Figure 4.2(3).7.3-6.

The quality of the zooplankton food is assigned in the User's Control Input by the parameter ZFOOD. Three qualities of food are allowed. From these, one type must be chosen to represent the overall food source available to the zooplankton:

ZFOOD = 1 = high quality food
2 = medium quality
3 = low quality

Depending on the value assigned to ZFOOD, the assimilation efficiency ZEFF is calculated by one of the following equations:

```
IF ZFOOD = 1 THEN ZEFF = -.06*PHYTO + 1.03                    (22)
   IF ZEFF > 0.99 THEN ZEFF = 0.99
```

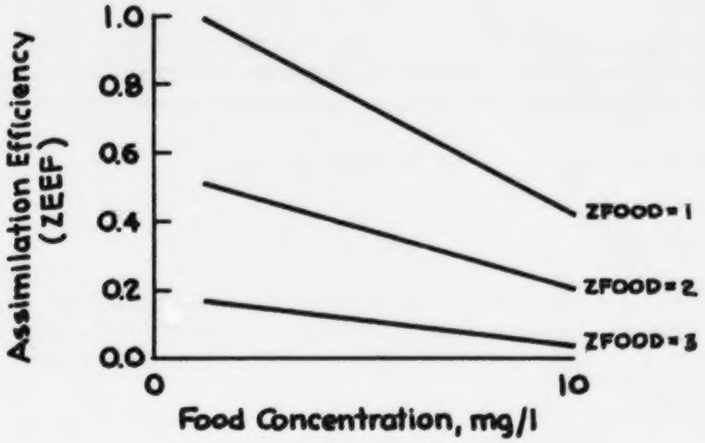


Figure 4.2(3).7.3-6 Zooplankton assimilation efficiency

```
IF ZFOOD = 2 THEN ZEFF = -.03*PHYTO + 0.47
   IF ZEFF < .20 THEN ZEFF = 0.20

IF ZFOOD = 3 THEN ZEFF = -.013*PHYTO + 0.17
   IF ZEFF < .03 THEN ZEFF = 0.03
```

These equations are extrapolations from research on Daphnia (Schindler 1968). The corrections to ZEFF set reasonable upper or lower limits on efficiency for assimilating each type of food. The mass of ingested phytoplankton assimilated by zooplankton is calculated as:

ZOGR = ZEFF*ZEAT (23)

where:

ZOGR = zooplankton growth as mg biomass/l per interval
ZEFF = assimilation efficiency (dimensionless)
ZEAT = ingested phytoplankton in mg biomass/l per interval

Respiration

Respiration is the biochemical process by which organic molecules are broken down, resulting in a release of energy which is essential for cellular and organismal activities. The oxidized molecules may either be carbohydrates and fats stored within the organism or food passing through the organism's digestive system. In either case, the end result of respiration is a decrease in zooplankton mass and a subsequent release of inorganic nutrients. The equation governing zooplankton respiration is:

ZRES = ZRES20*(TCZRES**(TW - 20.))*ZOO (24)

where:

ZRES = zooplankton biomass respired mg zoo/l per interval
ZRES20 = respiration rate at 20 degrees C, defaulted to .0015/hr
TCZRES = temperature correction factor for respiration, defaulted to 1.07
ZOO = zooplankton in mg biomass/l

Excretion Products

Excretion is the ingested food which is not assimilated by the zooplankton. These waste products contain both refractory and nonrefractory materials. The amount of refractory organic excretion is calculated as:

$$\mathrm{ZREFEX} = \mathrm{REFR} * \mathrm{ZEXMAS} \qquad (25)$$

where:

ZREFEX = refractory organic material excreted by zooplankton mg refractory biomass/l per interval
ZEXMAS = total mass of zooplankton excretion (ZEXMAS is the difference between ZEAT and ZOGR)
REFR = fraction of biomass which is refractory (REFR is the complement of parameter NONREF)

The nonrefractory portion of the excretion is released to the water in the form of inorganic nutrients and undegraded BOD materials. The relative abundance of the materials is dependent upon the unit ingestion rate of the zooplankton (ZOEAT). At higher ingestion rates, a larger fraction of the nonrefractory excretion is not decomposed and is released as BOD materials. In the model the parameter ZEXDEL is the fraction of nonrefractory excretion which is immediately decomposed and released to the water as inorganic nutrients when the unit ingestion rate of the zooplankton is maximal. If the unit ingestion rate is less than maximal, the model assumes that all the nonrefractory excretion is released to the water as inorganic nutrients. Thus, the amount of excretion released as inorganic materials is:

$$\mathrm{ZINGEX} = \mathrm{ZEXDEC} * (\mathrm{ZEXMAS} - \mathrm{ZREFEX}) \qquad (26)$$

where:

ZINGEX = amount of biomass decomposed to inorganic excretion as mg biomass/l per interval
ZEXDEC = fraction of nonrefractory inorganic excretion (ZEXDEC = 1 for ZOEAT <= MZOEAT and ZEXDEC = ZEXDEL for ZOEAT > MZOEAT. Value of ZOEAT is that given by equation 20; that is, prior to adjustment.)

The remaining portion of the excretion is considered to be BOD materials, and is calculated as:

$$\mathrm{ZNRFEX} = \mathrm{ZEXMAS} - \mathrm{ZREFEX} - \mathrm{ZINGEX} \qquad (27)$$

where:
ZNRFEX = amount of biomass released as nonrefractory organic excretion as mg biomass/l per interval

Death

Zooplankton death is the termination of all ingestion, assimilation, respiration, and excretion activities. After death, zooplankton contribute both refractory and nonrefractory materials to the system. Under aerobic conditions the mass rate of zooplankton death is determined by multiplying the natural zooplankton death rate, ZD, by the zooplankton concentration. If anaerobic conditions exist, an increase in zoopankton death rate is modeled by adding the value of the anaerobic death rate parameter, OXZD, to ZD. The default value of ZD is .0001/hr and that of OXZD is .03/hr.

Materials Balance for Related Constituents

Research has shown that 1.10 mg of oxygen are consumed for every gram of zooplankton mass which is respired (Richman 1958). The DOX state variable is reduced accordingly in ZORX. If there is not sufficient oxygen available to satisfy respiration requirements, the deficit is added to the BOD state variable, and DOX is set equal to zero.

ZORX makes use of subroutine DECBAL (4.2(3).7.2.4) to update the state variables NH3, NO3, and PO4 to account for additions from zooplankton respiration and inorganic excretion. The amount of inorganic constituents produced by these two processes is calculated by the following equations:

$$\begin{aligned} ZNIT &= (ZINGEX + ZRES)*CVBN \qquad (28) \\ ZPO4 &= (ZINGEX + ZRES)*CVBP \\ ZCO2 &= (ZINGEX + ZRES)*CVBC \end{aligned}$$

where:
ZNIT = increment to NH3 or NO3 state variable in mg N/l per interval
ZPO4 = increment to PO4 state variable in mg P/l per interval
ZCO2 = increment to CO2 state variable in mg C/l per interval
ZINGEX = amount of biomass decomposed to inorganic excretion expressed as mg biomass/l per interval
ZRES = amount of biomass respired by zooplankton as mg biomass/l per interval
CVBN = conversion factor from biomass to equivalent nitrogen
CVBP = conversion factor from biomass to equivalent phosphorus
CVBC = conversion factor from biomass to equivalent carbon

If ammonia is simulated, the inorganic nitrogen released is added to the NH3 variable; otherwise, it is added to the NO3 variable. The value of ZCO2 is calculated for use in subroutine group PHCARB if pH simulation is performed.

Finally, ZORX calls subroutine ORGBAL (4.2(3).7.3.3.4) to update the state variables for ORN, ORP, ORC, and BOD to account for additions from zooplankton death and organic excretion. The amount of organic constituents produced by these processes are calculated as:

```
 ORN = ((REFR*ZDTH) + ZREFEX)*CVBN                                 (29)
ZORP = ((REFR*ZDTH) + ZREFEX)*CVBP
ZORC = ((REFR*ZDTH) + ZREFEX)*CVBC
ZBOD = (ZDTH*CVNRBO) + (ZNRFEX*CVBO)
```

where:

ZORN = increment to ORN state variable in mg N/l per interval
ZORP = increment to ORP state variable in mg P/l per interval
ZORC = increment to ORC state variable in mg C/l per interval
ZBOD = increment to BOD state variable in mg O/l per interval
REFR = refractory fraction of biomass
ZDTH = zooplankton death as mg biomass/l per interval
ZREFEX = refractory organic excretion as mg biomass/l per interval
ZNRFEX = nonrefractory organic excretion as mg biomass/l per interval
CVBO = conversion from biomass to equivalent oxygen
CVNRBO = conversion from nonrefractory biomass to equivalent oxygen, times NONREF

4.2(3).7.3.5 Simulate Benthic Algae (subroutine BALRX)

Purpose

BALRX simulates those algae in the RCHRES which are attached to rocks or other stable structures. In free flowing streams, large diurnal fluctuations of oxygen can be attributed to benthic algae. During the sunlight hours, if sufficient nutrients exist to support photosynthesis, oxygen is produced in such large quantities that supersaturation often occurs. However, at night, when photosynthesis cannot occur, the benthic algae can exert a significant demand on the oxygen supply of the RCHRES due to respiratory requirements. Benthic algae influence the nutrient balance of the RCHRES by their extraction of nutrients for growth.

Approach

The growth and death of benthic algae are modeled in much the same manner as their free floating relatives, the phytoplankton. In fact, four of the five subroutines which are used for phytoplankton simulation are also used in the benthic algae simulation. These subroutines are ALGRO, GROCHK, ORGBAL, and NUTRUP. There are two major differences in modeling the two types of algae. First, since the benthic algae are attached to materials in the RCHRES, they are not subject to longitudinal advection. Second, the manner in which death of benthic algae is modeled is sufficiently different from the method used for phytoplankton that a special subroutine, BALDTH, is used. Within BALRX benthic algae are in units of micromoles phosphorus per liter so that the benthic algae simulation can take advantage of the same subroutines used by PHYRX. In order to obtain these units, the following conversion is performed:

BAL = BENAL*DEPCOR/CVPB (30)

where:

BAL = benthic algae as micromoles phosphorus/l
BENAL = benthic algae as mg biomass/m2
CVPB = conversion factor from micromoles phosphorus to mg biomass
DEPCOR = conversion from square meters to liters based on average depth of water in RCHRES during the interval (DEPCOR is calculated in RQUAL)

Net Growth

Unit growth and respiration rates for benthic algae are calculated by subroutine ALGRO. The user has the option of multiplying either of these rates by a constant factor if there is evidence that the benthic algae population does not exhibit the same growth and respiration rates as the phytoplankton population. Thus, net growth rate is calculated as:

GROBAL = (GRO*CFBALG - RES*CFBALR)*BAL (31)

where:

GROBAL = net growth rate of benthic algae as micromoles phosphorus/l per interval
GRO = unit growth rate as calculated in subroutine ALGRO
CFBALG = ratio of benthic algae to phytoplankton growth rates under identical growth conditions (default = 1.0)
RES = unit respiration rate as calculated in subroutine ALGRO
CFBALR = ratio of benthic algae to phytoplankton respiration rates (default = 1.0)
BAL = benthic algae concentration as micromoles phosphorus/l

After GROBAL has been calculated, subroutine GROCHK is called to assure that calculated growth does not reduce any nutrient to a concentration less than .001 mg/l. If it does, GROBAL is adjusted to satisfy this requirement.

Death of Benthic Algae

Subroutine BALDTH calculates the amount of benthic algae death and passes this information back to BALRX (variable DTHBAL). BALRX updates the state variable BAL to account for net growth and death. The value of BAL is not allowed to fall below .0001 micromoles of phosphorus per square meter.

Materials Balance for Related Constituents

The DOX state variable is updated to account for the net effect of benthic algae photosynthesis and respiration according to the following equation:

```
DOX = DOX + (CVPB*CVBO*GROBAL)                                    (32)
```

where:
DOX = concentration of dissolved oxygen in mg/l
CVPB = conversion factor from micromoles phosphorus to mg biomass
CVBO = conversion factor from mg biomass to mg oxygen
GROBAL = net growth of benthic algae as micromoles phosphorus/l per interval

The additions to ORN, ORP, ORC, and BOD resulting from benthic algae death are calculated as:

```
BALORN = REFR*DTHBAL*CVBPN*.014                                   (33)
BALORP = REFR*DTHBAL*.032
BALORC = REFR*DTHBAL*CVBPC*.012
BALBOD = CVNRBO*CVPB*DTHBAL
```

where:
BALORN = increment to ORN state variable in mg N/l per interval
BALORP = increment to ORP state variable in mg P/l per interval
BALORC = increment to ORC state variable in mg C/l per interval
BALBOD = increment to BOD state variable in mg O/l per interval
REFR = refractory fraction of biomass
DTHBAL = benthic algae death as micromoles P/l per interval
CVNRBO = conversion from mg biomass to equivalent mg oxygen demand (allowing for refractory fraction)
CVPB = conversion from micromoles phosphorus to mg biomass
CVBPN = conversion from micromoles phosphorus to micromoles nitrogen
CVBPC = conversion from micromoles phosphorus to micromoles carbon

When BALORN, BALORP, BALORC, and BALBOD have been evaluated, subroutine ORGBAL is called to perform the actual increments to the appropriate state variables. Finally, subroutine NUTRUP is called to update the inorganic state variables to account for net growth.

External Units

The output values for benthic algae are in units of milligrams biomass per square meter and micrograms chlorophyll a per square meter.

4.2(3).7.3.5.1 Calculate Benthic Algae Death (subroutine BALDTH)

Purpose

BALDTH calculates algal death each interval by using one of two unit death rates specified in the User's Control Input. ALDL, the low unit death rate, is used when environmental conditions encourage sustained life; in situations where nutrients are scarce or the benthic algae population becomes excessive, ALDH, the high algal death rate, is used.

Method

The high algal death rate, which has a default value of .01/hr, is used if any one of three conditions exists:

1. the concentration of PO4 is less than the value of parameter PALDH
2. the concentration of inorganic nitrogen is less than the value of parameter NALDH
3. the areal density of benthic algae is greater than the value of parameter MBAL

Regardless of whether these tests indicate that ALDH or ALDL (default equals .001/hr) should be used, an additional increment to death occurs if anaerobic conditions are prevalent during the interval. The increment to death rate due to anaerobic conditions is determined by the value of parameter OXALD. When the benthic algae population grows to a size greater than that which may be supported on the bottom surface, algae begin to break away from the bottom, a phenomenon known as sloughing. Whenever the population calculated exceeds the maximum allowable bottom density (MBAL), the sloughing process removes the excess algae. The amount of benthic algae death which occurs during the interval is calculated as:

$$DTHBAL = (ALD*BAL) + SLOF \qquad (34)$$

where:

DTHBAL = amount of benthic algae death as micromoles P/l per interval
ALD = unit algal death rate determined by environmental conditions in units of per interval
BAL = concentration of benthic algae as micromoles P/l
SLOF = amount of benthic algae sloughed as micromoles P/l per interval

4.2(3).7.4 Simulate pH, Carbon Dioxide, Total Inorganic Carbon, and Alkalinity (Subroutine Group PHCARB of Module RCHRES)

Purpose

PHCARB calculates the pH of the water within a RCHRES. The primary value of pH is as an indicator of the chemical environment of the system. Under normal circumstances, pH is near neutral, that is, near seven. Most life sustaining processes are impaired at extremes of pH.

Method

Figure 4.2(3).7.4-1 illustrates the fluxes and storages of constituents introduced in this section. Determination of pH requires simulation of alkalinity, carbon dioxide, and total inorganic carbon. Within PHCARB, state variables for alkalinity (ALK), carbon dioxide (CO2), and total inorganic carbon (TIC) are expressed as molar concentrations to correspond to the equilibrium expressions necessary to determine pH. The conversion from mg/l to moles/l takes place after longitudinal advection has been considered. Externally, ALK, CO2, and TIC are expressed in mg/l.

Alkalinity

Alkalinity is defined as the amount of acid required to attain a pH value equal to that of a total inorganic carbon molar solution of H_2CO_3. This pH value is near 4.5, which is approximately the lowest pH value tolerated by most forms of aquatic life. Alkalinity is interpreted as the acid neutralizing capacity of natural waters.

Alkalinity is simulated as a conservative constituent, in module section CONS. Parameter ALKCON, in the User's Control Input for PHCARB, specifies which conservative substance is alkalinity. For example, if ALKCON = 3 then subroutine PHCARB will assume that alkalinity is the 3rd conservative constituent.

Carbon Dioxide and Total Inorganic Carbon

HSPF assumes that changes in the TIC concentration occur only as changes in CO2 concentration. Thus, the sources of TIC are:

1. carbon dioxide invasion (input) from the atmosphere
2. zooplankton respiration
3. carbon dioxide released by BOD decay
4. net growth of algae (if negative)
5. benthal release of carbon dioxide (if BENRFG = 1)

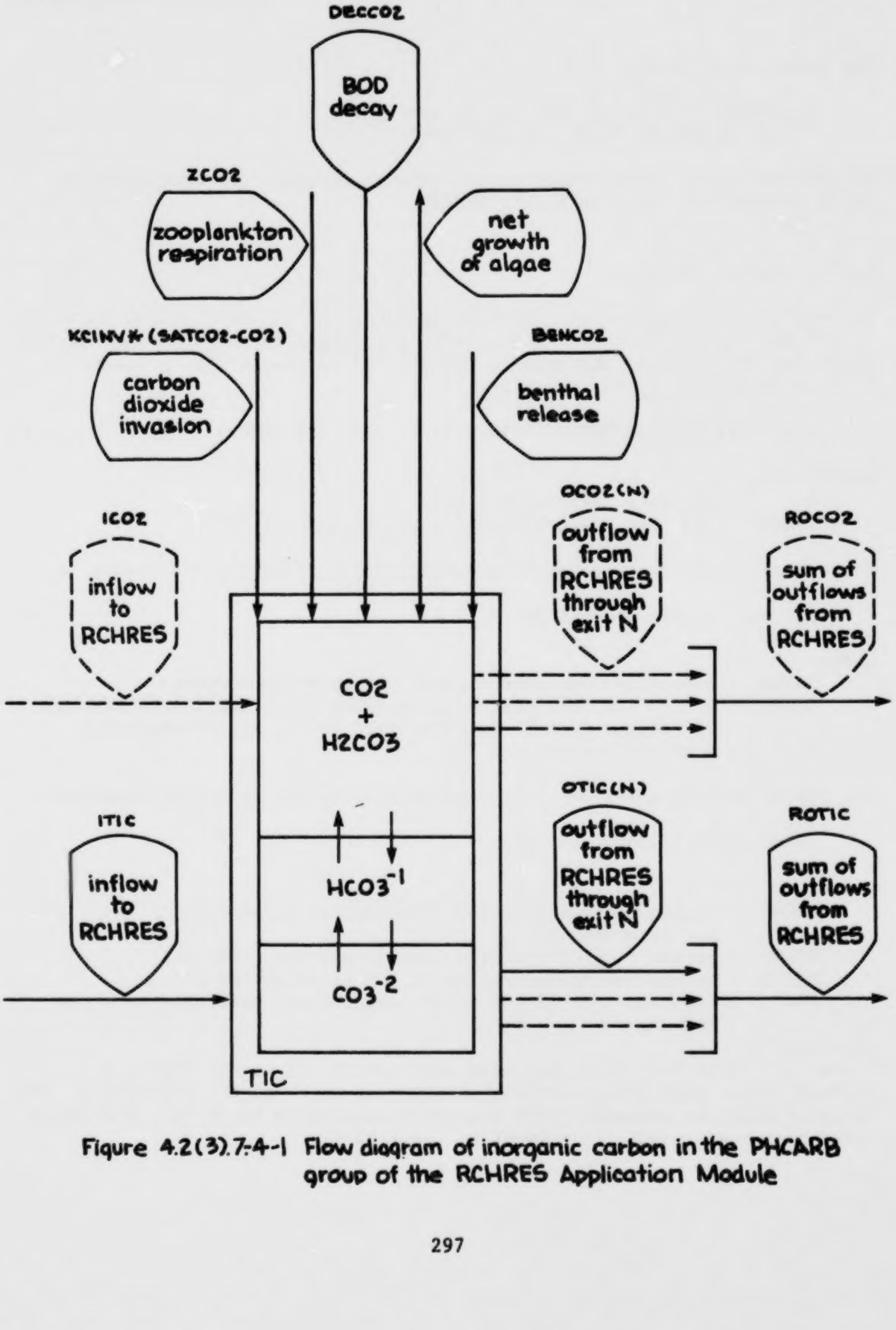


Figure 4.2(3).7-4-1 Flow diagram of inorganic carbon in the PHCARB group of the RCHRES Application Module

The sinks of TIC are:

1. carbon dioxide release to the atmosphere
2. net growth of algae (if positive)

All of these quantities except carbon dioxide invasion are calculated in other subroutines and passed into PHCARB.

Carbon Dioxide Invasion

In order to calculate carbon dioxide invasion, the saturation concentration of CO2 must be determined. First, Henry's constant for CO2, defined as the molar concentration of atmospheric CO2 divided by the partial pressure of CO2, is calculated by the equation:

```
S = 10.**(2385.73/TWKELV - 14.0184 + .0152642*TWKELV)          (1)
```

where:
S = Henrys's constant for CO2
TWKELV = absolute temperature of water in degrees Kelvin

Using Henry's constant, saturation concentration of CO2 is calculated as:

```
SATCO2 = 3.16E-04*CFPRES*S                                     (2)
```

where:
SATCO2 = saturation concentration of CO2 in moles CO2-C/l
CFPRES = correction to atmospheric pressure resulting from elevation difference (CFPRES is calculated in the Run Interpreter)
S = Henry's constant for CO2

The carbon dioxide invasion is then calculated by the following equation:

```
ATCO2 = KCINV*(SATCO2 - CO2)                                   (3)
```

where:
ATCO2 = carbon dioxide invasion expressed as moles CO2-C/l per interval
KCINV = carbon dioxide invasion coefficient (per interval)
SATCO2 = saturation concentration of CO2 in moles CO2-C/l
CO2 = concentration of CO2 after longitudinal advection in moles CO2-C/l

A positive value for ATCO2 indicates addition of CO2 to the water; a negative value indicates a release of CO2 from water to the atmosphere. The value of KCINV is dependent upon the value calculated for KOREA, the oxygen reaeration coefficient, in subroutine group OXRX:

$$\text{KCINV} = \text{CFCINV*KOREA} \tag{4}$$

where:
KCINV = carbon dioxide invasion coefficient (units are 1/interval)
CFCINV = parameter specifying ratio of CO2 invasion rate to O2 reaeration rate
KOREA = oxygen reaeration coefficient (units are 1/interval)

Net Carbon Dioxide Flux

The net carbon dioxide flux is determined by the following equation:

$$\text{DELTCD} = \text{ATCO2} + (\text{ZCO2} - \text{ALGCO2} + \text{DECCO2} + \text{BENCO2})/12000. \tag{5}$$

where:
DELTCD = net CO2 flux in moles CO2-C/l per interval
ATCO2 = CO2 invasion in moles CO2-C/l per interval
ZCO2 = CO2 released by zooplankton excretion and respiration in mg CO2-C/l per interval
ALGCO2 = CO2 flux due to net growth of algae in mg CO2-C/l per interval
DECCO2 = CO2 released by BOD decay in mg CO2-C/l per interval
BENCO2 = benthal release of CO2 in mg CO2-C/l per interval
12000. = conversion from mg CO2-C/l to moles CO2-C/l

If DECFG, the flag which decouples CO2 from algal simulation, has a value of one, ALGCO2 has a value of zero in this equation. Benthal release rates for both aerobic and anaerobic conditions must be included in the User's Control Input if benthal release of CO2 is simulated. Since HSPF assumes that changes in total inorganic carbon concentration only occur as changes in carbon dioxide, the update to the TIC state variable for each simulation interval is:

$$\text{TIC} = \text{TIC} + \text{DELTCD} \tag{6}$$

where:
TIC = total inorganic carbon in moles C/l

The Carbon System

The value of pH is controlled by the carbon system. There are three species of importance to the system: [H2CO3*], [HCO3], and [CO3]. [H2CO3*] is defined as the sum of [H2CO3] and [CO2]; for modeling purposes [H2CO3] is negligible relative to [CO2]. The carbon system can be described by the following equations:

$$\begin{aligned}
&[\text{H}]*[\text{HCO3}]/[\text{H2CO3*}] = \text{K1EQU} \\
&[\text{H}]*[\text{CO3}]/[\text{HCO3}] = \text{K2EQU} \\
&[\text{H}]*[\text{OH}] = \text{KWEQU}
\end{aligned} \tag{7}$$

[H2CO3*] + [HCO3] + [CO3] = TIC
[HCO3] + 2*[CO3] + [OH] - [H] = ALK

where:
[H] = hydrogen ion concentration in moles/l
[OH] = hydroxide ion concentration in moles/l
[CO3] = carbonate ion concentration in moles/l
[HCO3] = bicarbonate ion concentration in moles/l
[H2CO3*] = carbonic acid/carbon dioxide concentration in moles/l
K1EQU = first dissociation constant for carbonic acid
K2EQU = second dissociation constant for carbonic acid
KWEQU = ionization product of water

The five unknown values ([H2CO3*], [HCO3], [CO3], [H], [OH]) can be determined when K1EQU, K2EQU, KWEQU, TIC, and ALK are known. K1EQU, K2EQU, and KWEQU are all functions of water temperature and are evaluated by the following equations:

K1EQU = 10.**(-3404.71/TWKELV + 14.8435 - .032786*TWKELV) (8)
K2EQU = 10.**(-2902.39/TWKELV + 6.4980 - .02379*TWKELV)
KWEQU = 10.**(-4470.99/TWKELV + 6.0875 - .01706*TWKELV)

where:
TWKELV = absolute temperature of water in degrees Kelvin

Calculation of pH and CO2

Once values have been determined for K1EQU, K2EQU, KWEQU, TIC, and ALK, an equilibrium equation can be developed for hydrogen ion concentration ([H]). The five equations representing the carbon system (Equation 7) can be reduced to a fourth order polynomial expression:

[H]**4 + COEFF1*([H]**3) + COEFF2*([H]**2) + COEFF3*[H] + COEFF4 = 0 (9)

where:
COEFF1 = ALK + K1EQU
COEFF2 = -KWEQU + ALK*K1EQU + K1EQU*K2EQU - TIC*K1EQU
COEFF3 = -2.*K1EQU*K2EQU*TIC - K1EQU*KWEQU + ALK*K1EQU*K2EQU
COEFF4 = -K1EQU*K2EQU*KWEQU
[H] = hydrogen ion concentration in moles/l

The solution of this equation is performed by subroutine PHCALC. Based on the hydrogen ion concentration calculated in PHCALC, the concentration of CO2 is recalculated as:

CO2 = TIC/(1. + K1EQU/HPLUS + K1EQU*K2EQU/(HPLUS**2)) (10)

where:
CO2 = carbon dioxide concentration in moles C/l
TIC = total inorganic carbon concentration in moles C/l

K1EQU = first dissociation constant of carbonic acid
K2EQU = second dissociation constant of carbonic acid
HPLUS = hydrogen ion concentration in moles H/l

Finally, the units of TIC, CO2, and ALK are converted back to mg/l for use outside of PHCARB.

4.2(3).7.4.1 Calcuate pH (subroutine PHCALC)

PHCALC uses the Newton-Raphson method to solve the fourth order polynomial expression for the hydrogen ion concentration (Equation 9). The user specifies the maximum number of iterations performed by assigning a value to parameter PHCNT. PHCALC continues the iteration process until the solutions for pH concentration of two consecutive iterations differ by no more than one tenth of a pH unit. If the solution technique does not converge within the maximum allowable number of iterations, PHCALC passes this information back to PHCARB by assigning a value of zero to CONVFG. An error message is printed and then PHCALC is called again, to repeat the unsuccessful iteration process. This time, the "debug flag" (PHDBFG) is set ON so that, for each iteration, PHCALC will print information which will help the user or programmer to track down the source of the problem.

4.2(11) Copy Time Series (Utility Module COPY)

This utility module is used to copy one or more time series from a source specified in the EXT SOURCES or NETWORK Block of the User's Control Input (UCI), to a target specified in the NETWORK or EXT TARGETS Block (Part F, Section 4.6).

To operate the COPY module, the user must specify the time interval used in the internal scratch pad (INDELT) and the number of point-valued and mean-valued time series to be copied (NPT and NMN in Part F, Section 4.4(11).1). Up to 20 point-valued and/or 20 mean-valued time series may be copied in a single operation.

Module TSGET transfers the time series from the source(s), which may be either external (eg. TSS Dataset or sequential file) or the output(s) from one or more preceding operations, to the INPAD. If the time step changes it automatically aggregates or disaggregates. It also automatically alters the "kind" of time series, if appropriate, and can multiply each value by a user-specified factor.

Module TSPUT then transfers the time series from the INPAD to the target which, again, can be either external or internal. The work performed is a mirror image of that done by TSGET; time series can be aggregated/disaggregated and/or transformed in the same way.

Module COPY is typically used to transfer time series, such as precipitation and potential evapotranspiration data, from a sequential file (eg. card images) to a dataset in the Time Series Store (TSS). Thereafter, when these data are used as inputs to simulation operations, they are read directly from the TSS.

COPY can also be used to change the "kind" and/or interval of one or more time series. For example, a TSS dataset containing hourly precipitation data could be input to COPY and the output stored in another TSS dataset with a daily time step. The data would automatically be aggregated.

4.2(12) Prepare Time Series for Display on a Plotter (Utility Module PLTGEN)

This utility module prepares one or more time series for simultaneous display on a plotter. As with the COPY module (Section 4.2(11)), the user must specify the input(s) (sources), using entries in the EXT SOURCES or NETWORK Blocks in his control input (UCI). The internal time-step and the number of point- and/or mean-valued time series to be displayed must also be specified.

TSGET transfers the time series from the source(s) to the INPAD (as in COPY). PLTGEN then outputs these data to a plot file (PLOTFL). This is a sequential file; the first 25 records contain general information, such as the plot heading, number of curves to be plotted, scaling information, etc. Each subsequent record contains:

Cols	Contents
1 4	Identifier (first 4 characters of title)
7 22	Date/time
27 36	Value for curve 1, for this date/time
37 46	Value for curve 2, for this date/time
etc	(repeats until data for all curves are supplied)

The time resolution of the PLOTFL is that of the internal scratch pad; that is, the date/time field is incremented by INDELT for each succeeding record.

The contents of a sample PLOTFL are listed below. To keep the listing short, only the first five values have been included:

```
Plot HSPF FILE FOR DRIVING SEPARATE PLOT PROGRAM
Plot Time interval:   30 mins          Last month in printout year:  9

Plot No. of curves plotted:  Point-valued:  2   Mean-valued:  0   Total:  2


Plot Label flag:  0
Plot Plot title:   Plot of reservoir flowrates
Plot Y-axis label: Flow (ft3/sec)
Plot Scale info:  Ymin:   .00000E+00
Plot              Ymax:   1000.0
Plot              Time:   48.000     intervals/inch
Plot Data for each curve (Point-valued first, then mean-valued):
Plot Label                 LINTYP     INTEQ    COLCOD
Plot Inflow                       0          0          1
Plot Outflow                      0          0          1
Plot
Plot
Plot
Plot
Plot
Plot
Plot
Plot
Plot Time series (pt-valued, then mean-valued):
Plot
Plot Date/time                        Values
Plot
Plot  1974  5 31 24  0    .00000E+00     1.0000
Plot  1974  6  1  0 30    .82838         1.0000
Plot  1974  6  1  1  0    1.5071         1.0000
Plot  1974  6  1  1 30    2.0631         1.0000
Plot  1974  6  1  2  0    2.5186         1.0000
```

A plot file is intended to be read by a stand-alone plot program, which translates its contents into information used to drive a plotting device. Alternative uses of a PLOTFL are:

1. To display one or more time series in printed form. For example: To examine the contents of a dataset in the TSS, run it through PLTGEN and list the contents of PLOTFL on a line printer or terminal.

2. To feed time series to some other stand-alone program. For example, one could specify the contents of PLOTFL as input to a program which performs statistical analysis or computes cross correlations between time series.

4.2(13) Display Time Series in a Convenient Tabular Format (Utility Module DISPLY)

The purpose of this module is to permit any time series to be displayed (at a variety of time intervals) in a convenient format. Sample outputs are shown in Figures 4.2(13)-1 thru -3. Salient features of this module are:

1. Any time series (input or computed) can be displayed. The user specifies the time series in the EXT SOURCES or NETWORK Block, as with any other module.

2. As with any other module, the data are first placed in the INPAD, by module TSGET. At this point they are at the time interval specified for this operation in the OPN SEQUENCE Block (INDELT). This might have involved aggregation or disaggregation , if the data were brought in from the TSS. In general, INDELT can be any of the 19 HSPF supported time steps, ranging from 1 minute to 1 day.

3. The user can elect to display the data in a "long-span table" or a "short-span table". The term "span" refers to the period covered by each table. A short-span table (Figures 4.2(13)-1 and -2) covers a day or a month at a time and a long-span table (Figure 4.2(13)-3) covers a year.

4. The user selects the time-step for the individual items in a short-span display (the display interval) by specifying it as a multiple (PIVL) of INDELT. For example, the data in Figure 4.2(13)-1 are displayed at an interval of 5 minutes. This could have been achieved with:

INDELT	PIVL
5 min	1
1 min	5

If the display interval is less than an hour, an hours worth of data are displayed on one printed "row" (Figure 4.2(13)-1). The number of items in a row depends on their interval (eg. 60 for one minute, 12 for 5 minutes, 2 for 30 mins.). A "row" may actually occupy up to 5 physical lines of printout because a maximum of 12 items is placed on a line.

If the display interval is >= hour, a day's worth of data are displayed on one "row" (Figure 4.2(13)-2). Again, the number of items in a row depends on the display interval. In this case the entire table spans a month; in the former case it only spans a day.

5. A long-span table always covers a year; the display interval for the individual items in the table is a day (Figure 4.2(13)-3). The user can select the month which terminates the display (December, in the example) so that the data can be presented on a calendar year, water year or some other basis.

6. For the purpose of aggregating the data from the interval time step (INDELT) to the display interval, day-value, month-value, or year-value, one of five "transformation codes" can be specified:

Code	Meaning
SUM	Sum of the data
AVER	Average of the data
MAX	Take the max of the values at the smaller time step
MIN	Take the minimum
LAST	Take the last of the values belonging to the shorter time step

SUM is appropriate for displaying data like precip; AVER is useful for displaying data such as temperatures.

7. The module incorporates a feature designed to permit reduction of the quantity of printout produced when doing short-span displays. If the "row-value" ("hour-sum" in Figure 4.2(13)-1; "day-average" in Figure 4.3(13)-2) is less than or equal to a "threshold value", printout of the entire row is suppressed. The default threshold is 0.0. Thus, in Figure 4.2(13)-1; data for dry hours are not printed.

8. The user can also specify:

   a. The number of fractional digits to use in a display.

   b. A title for the display.

   c. A linear transformation, to be performed on the data when they are at the INDELT time interval (i.e. before module DISPLY performs any aggregation). By default, no transformation is performed.

```
                         TSS 2 Precip. (in/100)
                         Summary for DAY    1974/ 9/ 2
                         Data interval:     5 mins

HOUR    SUM              Interval Number.
                  1     2     3     4     5     6     7     8     9    10    11    12

  3     2.0      .0    .0    .0    .0    .0    .0    .0    .0    .0   1.0    .0   1.0
  4     3.0      .0    .0   1.0    .0    .0    .0   1.0    .0    .0   1.0    .0    .0
  5     5.0      .0    .0    .0    .0   1.0    .0    .0    .0    .0    .0   2.0   2.0
  6     6.0     1.0   1.0   2.0   1.0   1.0    .0    .0    .0    .0    .0    .0    .0
  7     3.0     1.0    .0    .0    .0    .0   1.0    .0    .0    .0    .0    .0   1.0
  8     3.0      .0    .0   1.0    .0    .0   1.0    .0    .0    .0   1.0    .0    .0
  9     3.0      .0    .0   1.0    .0    .0    .0    .0   1.0    .0    .0   1.0    .0
 10     3.0      .0    .0    .0    .0   1.0    .0    .0    .0   1.0    .0    .0   1.0
 11     3.0      .0    .0   1.0    .0    .0    .0    .0   1.0    .0    .0   1.0    .0
 12     4.0     1.0   1.0    .0    .0   1.0    .0    .0    .0    .0   1.0    .0    .0

 13     3.0      .0   1.0    .0    .0    .0   1.0    .0    .0    .0   1.0    .0    .0
 14     2.0      .0    .0    .0   1.0    .0    .0    .0    .0    .0    .0   1.0    .0
 15     4.0      .0    .0   1.0    .0   1.0    .0    .0   1.0    .0    .0   1.0    .0
 16     7.0      .0    .0   1.0    .0   1.0   1.0   1.0   1.0   1.0    .0   1.0    .0
 17     3.0     1.0    .0    .0    .0    .0   1.0    .0    .0    .0    .0   1.0    .0
 18     6.0     1.0    .0    .0    .0    .0   1.0    .0   1.0   1.0    .0   1.0   1.0
 19     5.0     1.0   1.0   1.0    .0    .0   1.0    .0    .0    .0   1.0    .0    .0
 20     3.0      .0    .0    .0    .0    .0   1.0    .0    .0    .0   1.0    .0   1.0
 21     1.0      .0    .0    .0    .0    .0    .0    .0    .0   1.0    .0    .0    .0
 22     1.0      .0    .0    .0    .0    .0   1.0    .0    .0    .0    .0    .0    .0

DAY    SUM : 7.00000E+01
```

Figure 4.2(13)-1 Sample short-span display (first type)

```
                               TSS 3 Temperature (Deg F)
                               Summary for MONTH  1974/ 8/
                               Data interval:   120 mins

DAY      AVER                  Interval Number.
                  1     2     3     4     5     6     7     8     9    10    11    12

  1      63.8    54.5  53.5  52.5  53.0  59.5  68.0  74.5  76.0  74.5  71.0  66.0  62.5
  2      68.8    61.0  60.0  59.0  60.0  65.0  72.5  77.5  79.0  77.5  75.0  71.0  67.5
  3      68.6    65.5  65.0  64.0  64.5  68.5  73.5  77.0  78.0  76.0  70.5  63.5  57.0
  4      64.0    54.5  53.5  52.5  53.5  60.0  69.0  75.5  77.0  75.0  71.0  65.5  60.5
  5      64.9    58.5  57.0  56.0  57.0  62.0  69.5  74.5  76.0  74.0  70.0  64.5  59.5

  6      66.7    57.5  56.0  55.0  56.5  63.5  73.5  80.0  82.0  79.5  73.5  65.0  58.5
  7      66.6    55.5  53.5  52.5  53.5  61.0  71.5  79.0  81.0  79.5  75.5  70.5  66.0
  8      70.3    64.0  63.0  62.0  63.0  68.0  75.0  79.5  81.0  79.0  75.0  69.5  64.5
  9      68.7    62.5  61.0  60.0  61.0  66.0  73.5  78.5  80.0  78.0  73.5  67.5  63.0
 10      69.6    60.5  59.0  58.0  59.0  65.0  74.0  79.5  81.0  79.5  77.0  73.0  70.0

 11      72.8    68.5  68.0  67.0  67.5  71.5  77.5  81.0  82.0  80.0  75.5  69.5  65.0
 12      70.8    62.5  61.0  60.0  61.0  67.0  76.0  81.5  83.0  81.5  77.5  71.5  67.0
 13      70.3    65.5  64.0  63.0  64.0  69.0  76.0  80.5  82.0  80.0  74.0  66.0  60.0
 14      65.5    57.5  55.5  54.5  55.5  62.0  71.0  77.5  79.0  77.0  72.0  65.0  59.0
 15      67.1    56.5  55.5  54.5  55.5  62.5  72.5  79.0  81.0  79.0  75.0  69.5  64.5

 16      70.1    62.5  61.0  60.0  61.0  67.0  75.0  80.5  82.0  80.0  76.0  70.5  65.5
 17      66.8    63.5  62.0  61.0  61.5  65.5  71.5  75.0  76.0  74.0  69.5  63.5  58.0
 18      66.2    55.5  54.0  53.0  54.5  61.5  72.5  79.0  81.0  79.0  74.5  67.5  62.0
 19      70.3    59.5  58.0  57.0  58.5  65.5  76.5  83.0  85.0  83.0  78.5  72.5  67.0
 20      73.8    64.5  63.0  62.0  63.5  70.0  80.0  86.0  88.0  86.0  81.0  74.0  68.0

 21      74.7    65.5  64.5  63.5  64.5  71.0  81.0  87.0  89.0  87.0  81.5  74.0  68.0
 22      73.3    65.0  63.5  62.5  63.0  69.5  78.0  84.5  86.0  84.0  80.0  74.5  69.5
 23      73.4    67.5  66.0  65.0  66.0  71.5  79.5  84.5  86.0  84.0  78.0  70.0  63.0
 24      66.2    60.5  58.5  57.5  58.0  64.0  73.0  78.5  80.0  77.0  71.0  62.0  54.5
 25      64.0    51.5  49.5  48.5  50.0  58.0  70.0  78.0  80.0  78.0  73.5  67.5  63.0

 26      72.9    60.5  59.0  58.0  59.5  67.5  79.5  87.0  89.0  87.0  82.5  75.5  70.0
 27      73.8    67.5  66.0  65.0  66.0  72.5  81.0  87.5  89.0  85.5  78.5  67.5  59.0
 28      60.3    55.0  53.0  51.5  52.0  57.0  64.5  69.5  71.0  69.5  65.5  59.5  55.0
 29      62.7    53.5  52.0  51.0  52.0  58.5  67.0  73.5  75.0  73.5  70.0  65.0  61.0
 30      66.9    59.0  58.0  57.0  58.0  63.5  71.5  76.5  78.0  76.5  73.0  68.0  64.0
 31      67.0    62.0  61.0  60.0  61.0  66.0  73.0  77.5  79.0  76.5  70.5  62.0  55.5

MONTH  AVER: 6.84059E+01
```

Figure 4.2(13)-2 Sample short-span display (second type)

```
                         TSS 3 Temperature (Deg F)
               Annual data display: Summary for period ending 1974/12
Day      JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG    SEP    OCT    NOV    DEC

  1     13.6   21.0   39.8   38.9   49.6   58.9   70.3   63.8   57.9   40.4   63.5   28.7
  2      9.1   15.8   42.2   49.5   51.2   59.9   76.0   68.8   54.9   33.4   55.7   29.8
  3     16.0   14.4   54.7   53.5   50.0   61.2   79.8   68.6   51.7   39.7   54.2   27.5
  4     15.8   12.1   52.2   51.0   45.8   69.6   75.8   64.0   51.8   52.0   42.0   20.7
  5     13.8    6.6   42.5   39.9   45.0   71.0   63.2   64.9   53.5   60.8   36.4   22.5
  6     16.6   13.0   48.3   36.9   38.3   72.2   67.3   66.7   56.1   60.3   37.5   24.6
  7     13.9    8.0   47.8   40.5   39.4   72.4   72.2   66.6   58.6   48.5   39.7   31.0
  8      5.0    6.9   42.4   32.8   44.5   73.0   77.6   70.3   66.0   45.1   43.4   28.7
  9     15.4   14.7   40.6   30.7   40.9   75.7   79.0   68.7   68.2   49.1   44.3   21.3
 10     15.8   16.2   40.1   42.3   47.5   71.2   75.6   69.6   69.8   51.3   46.4   25.5
 11     15.2   17.4   34.4   50.8   50.8   54.9   63.6   72.8   73.5   57.2   54.1   29.4
 12     11.0   31.8   34.3   56.2   50.5   57.1   64.6   70.8   72.7   58.2   40.4   34.0
 13     12.6   31.7   25.2   58.4   43.3   60.6   75.3   70.3   66.1   45.2   33.6   33.6
 14     25.7   17.7   29.8   55.3   57.3   68.4   80.9   65.5   52.4   53.5   27.2   32.3
 15     29.3   13.9   36.2   40.7   58.2   64.5   75.8   67.1   57.0   44.0   26.6   32.8
 16     30.5   23.1   35.5   41.7   52.8   58.8   66.7   70.1   53.8   45.5   31.2   32.2
 17     30.8   24.4   35.5   43.1   60.0   52.3   70.2   66.8   61.4   48.1   41.5   29.0
 18     27.5   26.7   33.8   47.3   55.8   60.4   77.0   66.2   59.0   38.2   45.3   21.2
 19     31.0   33.4   33.2   37.8   57.7   66.5   73.5   70.3   62.7   31.2   44.3   22.7
 20     35.0   31.9   28.7   47.5   57.8   71.1   66.9   73.8   64.7   30.2   41.3   24.8
 21     39.6   36.0   31.1   59.5   67.1   72.3   62.9   74.7   52.8   36.3   35.3   24.8
 22     33.8   34.1   30.2   58.3   71.0   64.4   67.9   73.3   46.1   51.6   36.0   25.4
 23     32.0   20.9   27.6   42.1   61.3   55.0   63.8   73.4   43.6   54.7   45.1   33.4
 24     32.3   13.0   12.3   39.9   56.2   57.0   67.8   66.2   50.9   49.8   44.0   32.3
 25     37.0   16.1   20.3   43.8   52.8   60.4   69.8   64.0   55.3   51.6   27.4   26.0
 26     40.5   24.7   33.2   54.1   51.9   64.3   73.8   72.9   58.7   46.3   21.7   24.4
 27     41.5   36.5   30.9   65.0   50.4   63.5   73.7   73.8   63.8   50.4   26.5   28.3
 28     35.2   43.9   31.4   68.2   55.7   64.3   70.1   60.2   64.5   56.5   28.0   31.0
 29     30.0          30.1   64.4   65.2   66.5   72.2   62.7   57.1   58.4   29.2   33.5
 30     37.6          34.6   60.1   65.1   70.1   65.8   66.9   45.2   64.3   25.7   32.6
 31     35.7          35.6          63.6          64.5   67.0          66.4          30.5

AVER    25.1   21.6   35.3   48.3   53.4   64.6   71.1   68.4   58.3   49.0   38.9   28.2

AVER of monthly values  4.68582E+01
```

Figure 4.2(13)-3 Sample long-span (annual) display

4.2(14) Perform Duration Analysis on a Time Series (Utility Module DURANL)

This module examines the behavior of a time series, computing a variety of statistics relating to its excursions above and below certain specified "levels" (Figure 4.2(14)-1). Sample printout is shown in Figure 4.2(14)-2. The quantity of printout produced can be regulated by the user with a "print-level-flag" (PRFG), which has a valid range of values from 1 through 6.

The basic principles are:

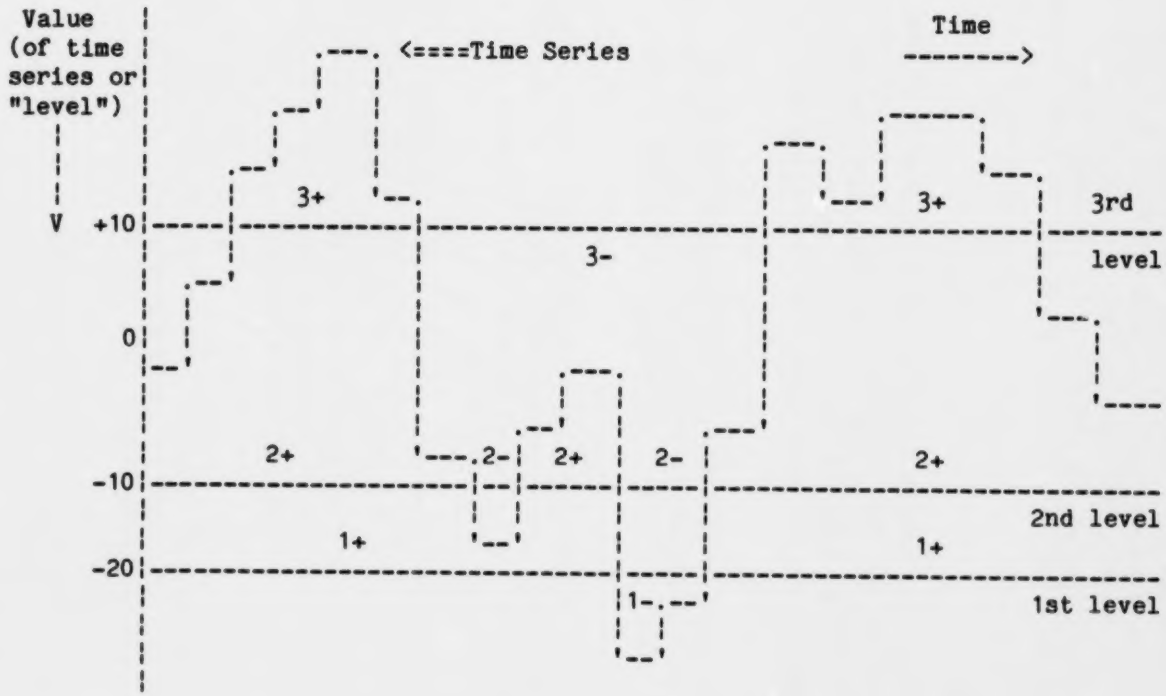
1. The module works on the time series after it has been placed in the INPAD. The data are, thus, at the internal time step of the operation (INDELT). This module operates on a mean-valued input time series. Therefore, if a point-valued time series is routed to it, TSGET will, by default, generate mean values for each time step, and these will be analyzed.

2. When the value of the time series rises above the user specified "level", a positive excursion commences. When it next falls below the level this excursion ends. A negative excursion is defined in the reverse way. (Figure 4.2(14)-1).

3. If the time series has a value less than -10.0**10 this is considered to be an "undefined event" (eg. concentration of a constituent when there is no water). In this case the value is in a special category - it is in neither a positive nor a negative excursion.

4. The above is true if the specified "duration" is one time step. In this case, the results produced include a conventional frequency analysis (eg. flow duration) of the data. However, the user may specify up to 10 durations; each is given as a multiple (N) of the basic time step (INDELT), Then, for an excursion or undefined event to be considered, it has to endure for at least N (consecutive) intervals; else it is ignored.

5. The user may specify an "analysis season". This is a period (the same in each year) for which the data will be analyzed (eg. Oct 1 thru May 10). Data falling outside the analysis season will not be considered.



Legend: 2+ excursion above second level (duration >=1)
2- excursion below second level (duration >=1)
etc.

Figure 4.2(14)-1 Definition of terms used in duration analysis module

The analyses performed, and printout produced (Figure 4.2(14)-2), are:

1. Introductory information - Title, start and end date/time, analysis season.

2. The next 6 sets of tables are all similar in format; each contains data on positive and negative excursions, for each level and duration, and information on "undefined event" conditions which persisted for each of the specified durations. The value of PRFG required to generate each of these, and the table heading and the data displayed in it are:

   a) PRFG>0. "Percent of Time Tables (with respect to the total span of time). These are the fractions of total considered time that each of the above- defined conditions existed.

   b) PRFG>1. "Percent of Time Tables (with respect to the time spent in excursions). In the "Positive Excursions" table, this gives, for each given level, the total time that an excursion of duration N existed, divided by the total time that an excursion of duration 1 existed. A similar definition holds for the numbers in the "Negative Excursions" table.

   c) PRFG>2. "Time Spent in Excursions". The tables give the total number of time steps for which the various conditions occurred.

   d) PRFG>3. "Frequency Tables". These give the total number of events that were found (no. of positive and negative excursions for each level and duration, and no. of "undefined occurences" of each duration).

   e) PRFG>4. "Expected Length of Time Spent in Excursions". These values answer the question: "given that a specified excursion or 'undefined condition' occurred, what was the mean number of time steps for which it persisted?"

   f) PRFG>5. "Standard Deviation of Time Spent in Excursions". These tables are similar to those discussed in (e) above, except that the standard deviation, instead of the mean, is considered.

3. Summary information:

   Total no. of time steps analyzed, total no. of time steps for which values were "undefined", sample size, max, min, mean, standard deviation.

```
Duration analysis operation no.      1
Analysis of Subb. 4 Outflow (cfs)
Start date: 1972/12/31 24: 0  End date: 1974/12/31 24: 0
Analysis season starts:  2/28 24: 0  Ends: 11/30 24: 0


PERCENT OF TIME TABLES (WITH RESPECT TO THE TOTAL SPAN OF TIME)

POSITIVE EXCURSIONS
                      DURATIONS
                     1          12          24
LEVELS
  .0000E+00  1.000       1.000       1.000
  10.00      .7308       .7259       .7235
  20.00      .5128       .5062       .5034
  50.00      .1790       .1674       .1633
  500.0      .2273E-02   .0000E+00   .0000E+00


NEGATIVE EXCURSIONS
                      DURATIONS
                     1          12          24
LEVELS
  .0000E+00  .0000E+00   .0000E+00   .0000E+00
  10.00      .2692       .2655       .2645
  20.00      .4872       .4813       .4762
  50.00      .8210       .8121       .8030
  500.0      .9977       .9977       .9977


UNDEFINED EVENTS (NO WATER)
                      DURATIONS
                     1          12          24
             .0000E+00   .0000E+00   .0000E+00
```

Figure 4.2(14)-2 Sample Duration Analysis Printout
[Continued on next 3 pages]

```
PERCENT OF TIME TABLES (WITH RESPECT TO THE TIME SPENT IN EXCURSIONS)

POSITIVE EXCURSIONS
                          DURATIONS
                         1          12          24
LEVELS
   .0000E+00  1.000       1.000      1.000
   10.00      1.000       .9933      .9899
   20.00      1.000       .9871      .9817
   50.00      1.000       .9353      .9124
   500.0      1.000       .0000E+00  .0000E+00

NEGATIVE EXCURSIONS
                          DURATIONS
                         1          12          24
LEVELS
   .0000E+00  .0000E+00  .0000E+00  .0000E+00
   10.00      1.000       .9862      .9828
   20.00      1.000       .9879      .9775
   50.00      1.000       .9892      .9780
   500.0      1.000       1.000      1.000

UNDEFINED EVENTS (NO WATER)
                          DURATIONS
                         1          12          24
              .0000E+00  .0000E+00  .0000E+00

TIME SPENT IN EXCURSIONS

POSITIVE EXCURSIONS
                          DURATIONS
                         1          12          24
LEVELS
   .0000E+00  .1320E+05  .1320E+05  .1320E+05
   10.00      9647.      9582.      9550.
   20.00      6769.      6682.      6645.
   50.00      2363.      2210.      2156.
   500.0      30.00      .0000E+00  .0000E+00

NEGATIVE EXCURSIONS
                          DURATIONS
                         1          12          24
LEVELS
   .0000E+00  .0000E+00  .0000E+00  .0000E+00
   10.00      3553.      3504.      3492.
   20.00      6431.      6353.      6286.
   50.00      .1084E+05  .1072E+05  .1060E+05
   500.0      .1317E+05  .1317E+05  .1317E+05

UNDEFINED EVENTS (NO WATER)
                          DURATIONS
                         1          12          24
              .0000E+00  .0000E+00  .0000E+00
```

```
FREQUENCY TABLES

POSITIVE EXCURSIONS
                      DURATIONS
                     1          12          24
LEVELS
  .0000E+00  1.000       1.000       1.000
  10.00      47.00       7.000       5.000
  20.00      77.00       18.00       16.00
  50.00      104.0       15.00       12.00
  500.0      14.00       .0000E+00  .0000E+00

NEGATIVE EXCURSIONS
                      DURATIONS
                     1          12          24
LEVELS
  .0000E+00  .0000E+00  .0000E+00  .0000E+00
  10.00      47.00       26.00       25.00
  20.00      77.00       47.00       43.00
  50.00      105.0       65.00       58.00
  500.0      15.00       15.00       15.00

UNDEFINED EVENTS (NO WATER)
                      DURATIONS
                     1          12          24
             .0000E+00  .0000E+00  .0000E+00

EXPECTED LENGTH OF TIME SPENT IN EXCURSIONS

POSITIVE EXCURSIONS
                      DURATIONS
                     1          12          24
LEVELS
  .0000E+00  .1320E+05  .1320E+05  .1320E+05
  10.00      205.3       1369.       1910.
  20.00      87.91       371.2       415.3
  50.00      22.72       147.3       179.7
  500.0      2.143       .0000E+00  .0000E+00

NEGATIVE EXCURSIONS
                      DURATIONS
                     1          12          24
LEVELS
  .0000E+00  .0000E+00  .0000E+00  .0000E+00
  10.00      75.60       134.8       139.7
  20.00      83.52       135.2       146.2
  50.00      103.2       164.9       182.7
  500.0      878.0       878.0       878.0

UNDEFINED EVENTS (NO WATER)
                      DURATIONS
                     1          12          24
             .0000E+00  .0000E+00  .0000E+00
```

```
STANDARD DEVIATION OF TIME SPENT IN EXCURSIONS


POSITIVE EXCURSIONS
                         DURATIONS
                       1          12          24
LEVELS
  .0000E+00  .0000E+00  .0000E+00  .0000E+00
  10.00      922.9      2032.      2181.
  20.00      321.6      581.1      602.0
  50.00      71.65      132.1      128.7
  500.0      .7423      .0000E+00  .0000E+00



NEGATIVE EXCURSIONS
                         DURATIONS
                       1          12          24
LEVELS
  .0000E+00  .0000E+00  .0000E+00  .0000E+00
  10.00      107.2      113.8      113.3
  20.00      127.0      140.0      141.4
  50.00      167.6      188.1      191.6
  500.0      1202.      1202.      1202.



UNDEFINED EVENTS (NO WATER)
                         DURATIONS
                       1          12          24
             .0000E+00  .0000E+00  .0000E+00



    SUMMARY
    TOTAL SPAN OF TIME:    13200.0
    TOTAL LENGTH OF UNDEFINED EVENTS:          .0
    SAMPLE SIZE:    13200
    SAMPLE MAXIMUM:  .1307E+05
    SAMPLE MINIMUM:  2.290
    SAMPLE MEAN:  37.80
    SAMPLE STANDARD DEVIATION:  164.0
```

4.2(15) Generate a Time Series from One or Two Other Time Series (Utility Module GENER)

This module is designed to perform any one of several possible transformations on input time series. The transformation is specified by supplying an "option code" (OPCODE). If A and B are the input time series and C is the computed time series, then the transformations performed for each possible value of OPCODE are:

OPCODE	Action
1	C= Abs value (A)
2	C= Square root (A)
3	C= Truncation (A) eg. If A=4.2, C=4.0 A=-3.5, C=-3.0
4	C= Ceiling (A). The "ceiling" is the integer >= given value. eg. If A=3.5, C=4.0 A=-2.0, C=-2.0
5	C= Floor (A). The "floor" is the integer <= given value. eg. If A=3.0, C=3.0 A=-2.7, C=-3.0
6	C= loge (A)
7	C= log10 (A)
8	C= K(1)+K(2)*A+K(3)*A**2 (up to 7 terms) The user supplies the no. of terms and the values of the coefficients (K).
9 thru 15	Spare, for future use.
16	C= A+B
17	C= A-B
18	C= A*B
19	C= A/B
20	C= MAX (A,B)
21	C= MIN (A,B)

Note that if OPCODE is <=8, only one input time series is involved (unary operators) but the other types of operations involve two inputs (binary operators). As with the other operating modules, the input time series are first placed in the INPAD by module TSGET (This may involve a change of time step and/ or "kind"). So, by the time module GENER works on them, they are mean valued time series with a time step equal to INDELT.

4.3 Module TSPUT

Module TSPUT is complementary to, and may be viewed as a mirror image of, module TSGET (Section 4.1). TSGET obtains time series from a TSS, sequential file or the INPAD and places its output in the INPAD. Conversely, TSPUT obtains a time series from the INPAD and places its output in the TSS or back in the INPAD. It has similar capabilities to TSGET, to alter the time step, "kind" or to perform a linear transformation on the time series with which it deals.

Compared to TSGET, module TSPUT contains one major complicating factor. When a time series is to be written to a TSS dataset, the action taken depends on how any pre-existing data are to be treated. The three possible access modes, ADD, INST and REPL, are discussed in Part F, Section 4.6.

REFERENCES

Anderson, E.A. 1968. Development and Testing of Snow Pack Energy Balance Equations. Water Resour. Res. 4(1):19-37.

Anderson, E.A., and N.H. Crawford. 1964. The Synthesis of Continuous Snowmelt Runoff Hydrographs on a Digital Computer. Department of Civil Engineering, Stanford University. Stanford, California. Technical Report No. 36. 103 p.

Churchill, M.A., H.L. Elmore, and R.A. Buckingham. 1962. The Prediction of Stream Reaeration Rates. Amer. Soc. Civil Engineers Journ. 88(SA4), p. 1-46.

Committee on Sanitary Engineering Research. 1960. Solubility of Atmospheric Oxygen in Water. Twenty-ninth Progress Report. Amer. Soc. Civil Engr., J. San. Engr. Div. 86(SA4):41.

Covar, A.P. 1976. Selecting the Proper Reaeration Coefficient for Use in Water Quality Models. Proc. Conf. on Environmental Modeling, and Simulation, Cincinnati. EPA 600/9-76-016. 861p.

Crawford, N.H., and A.S. Donigian, Jr. 1973. Pesticide Transport and Runoff Model for Agricultural Lands. Office of Research and Development, U.S. Environmental Protection Agency, Washington D.C. EPA 660/2-74-013. 211p.

Di Toro, D.M., D.T. O'Connor, and R.V. Thomann. 1970. A Dynamic Model of Phytoplankton Populations in Natural Waters. Environmental Engineering and Science Program. Manhattan College, New York.

Donigian, A.S., Jr., and N.H. Crawford. 1976a. Modeling Pesticides and Nutrients on Agricultural Lands. Environmental Research Laboratory, Athens, Georgia. EPA 600/2-7-76-043. 317 p.

Donigian, A.S., Jr., and N.H. Crawford. 1976b. Modeling Nonpoint Pollution From the Land Surface. Environmental Research Laboratory, Athens, Georgia. EPA 600/3-76-083. 280p.

Donigian, A.S, Jr., D.C. Beyerlein, H.H. Davis, Jr., and N.H. Crawford. 1977. Agricultural Runoff Management (ARM) Model Version II: Refinement and Testing. Environmental Research Laboratory, Athens, Georgia. EPA 600/3-77-098. 294p.

Dugdale, R.C., and J.J. Macisaac. 1971. A Computational Model for the Uptake of Nitrate in the Peru Upwelling. Prepublication Copy.

Harnard, H.S. and R. Davis. 1943. The Ionization Constant of Carbonic Acid in Water and the Solubiltiy of CO2 in Water and Aqueous Salt Solution from 0 to 50 C. J. Am. Chem. Soc. 65:2030.

Hydrocomp, Inc. 1977. Hydrocomp Water Quality Operations Manual. Palo Alto, California.

Hydrocomp, Inc. 1976. Hydrocomp Simulation Programming: Operations Manual. Palo Alto, California, 2nd ed.

Johanson, R,C., J.C. Imhoff and H.H. Davis, Jr. 1979. Programmer's Supplement for the Hydrological Simulation Program - Fortran (HSPF). This material is on magnetic tape. See Appendix IV.

Meyer, L.D., and W.H. Wischmeier. 1969. Mathematical Simulation of the Process of Soil Erosion by Water. Trans. Am. Soc. Agric. Eng. 12(6):754-758,762.

Negev, M. 1967. A Sediment Model on a Digital Computer. Department of Civil Engineering, Stanford University. Stanford, California. Technical Report No. 76. 109p.

O'Connor, D.J., and D.M. Di Toro. 1970. Photosynthesis and Oxygen Balance in Streams. Amer. Soc. Civil Engr., J. San. Engr. Div. 96(SA2):7240.

O'Connor, D.J., and W.E. Dobbins. 1958. Mechanism of Reaeration in Natural Streams. Amer. Soc. Civil Engineers Trans., Vol. 123, p. 641-684.

Onstad, C.A., and G.R. Foster. 1975. Erosion Modeling on a Watershed. Trans. Am. Soc. Agric. Eng. 18(2):288-292.

Owens, M., R.W. Edwards, and J.W. Gibbs. 1964.. Some Reaearation Studies in Streams. Int. Journ. Air and Water Pollution. Vol. 8, p. 469-486.

Philip, J.R. 1956. The Theory of Infiltration: The Infiltration Equation and Its Solution. Soil Science 83: 345-375.

Richman, S. 1958. The Transformation of Energy by Daphnia pulex. Ecolog. Monogr. Vol. 28, p. 273-291.

Schindler, D.W. 1968. Feeding, Assimilation and Respiration Rates of Daphnia magna Under Various Envirionmental Conditions and their Relation to Production Estimates. Journal of Animal Ecology. Vol. 37, p. 369-385.

Thomann, R.V. 1972. Systems Analysis and Water Quality Management. McGraw-Hill, Inc., New York. 286p.

Tsivoglou, E.C., and J.R. Wallace. 1972. Characterization of Stream Reaeration Capacity. U.S. Environmental Protection Agency, Rept. R3-72-012.

U.S. Army Corps of Engineers. 1956. Snow Hydrology, Summary Report of the Snow Investigations. North Pacific Division. Portland Oregon. 437 p.

U.S. Environmental Protection Agency. 1975. Process Design Manual for Nitrogen Control. Office of Technology Transfer, Washington D.C.

Wezerak, C.G., and J.J. Gannon. 1968. Evaluation of Nitrification in Streams. Amer. Soc. Civil Engr., J. San. Engr. Div. 94(SA5):6159.

Wischmeier, W.H., and D.D. Smith. 1965. Predicting Rainfall Erosion Losses from Cropland East of the Rocky Mountains. Department of Agriculture. Agricultural Handbook No. 282. 47 p.

PART F

FORMAT FOR THE USER'S CONTROL INPUT

CONTENTS

1.0 General Information and Conventions 322
    1.1 The Users Control Input. 322
    1.2 General Comments on Method of Documentation 322

2.0 Format of a TSSMGR Data Set 323
    2.1 Layout 323
    2.2 Details. 325
    2.3 Explanation. 326

3.0 Sample TSSMGR Data Set. 327

4.0 Format of a RUN Data Set. 328
    4.1 Summary. 328
    4.2 GLOBAL Block 329
    4.3 OPN SEQUENCE Block 331
    4.4 <Operation-type> Block 333
        4.4(1) PERLND Block 335
            4.4(1).1 General input 336
            4.4(1).2 Section ATEMP input 341
            4.4(1).3 Section SNOW input 343
            4.4(1).4 Section PWATER input 351
            4.4(1).5 Section SEDMNT input 367
            4.4(1).6 Section PSTEMP input 375
            4.4(1).7 Section PWTGAS input 386
            4.4(1).8 Section PQUAL input 395
            4.4(1).9 Section MSTLAY input 407
            4.4(1).10 Section PEST input 416
            4.4(1).11 Section NITR input 429
            4.4(1).12 Section PHOS input 441
            4.4(1).13 Section TRACER input 451
        4.4(2) IMPLND Block 455
            4.4(2).1 General input 455
            4.4(2).2 Section ATEMP input 460
            4.4(2).3 Section SNOW input 460
            4.4(2).4 Section IWATER input 460
            4.4(2).5 Section SOLIDS input 468
            4.4(2).6 Section IWTGAS input 474
            4.4(2).7 Section IQUAL input 480
        4.4(3) RCHRES Block 485
            4.4(3).1 General input 485
            4.4(3).2 Section HYDR input 490

4.4(3).3 Section ADCALC input 496
4.4(3).4 Section CONS input 498
4.4(3).5 Section HTRCH input 501
4.4(3).6 Section SED input 504
4.4(3).7 Section FSTORD input 507
4.4(3).8 Input for RQUAL sections 510
4.4(3).8.1 Section OXRX input 513
4.4(3).8.2 Section NUTRX input 524
4.4(3).8.3 Section PLANK input 530
4.4(3).8.4 Section PHCARB input 542
4.4(11) COPY Block 546
4.4(12) PLTGEN Block 548
4.4(13) DISPLY Block 554
4.4(14) DURANL Block 559
4.4(15) GENER Block. 566
4.5 FTABLES Block. 571
4.6 EXT SOURCES, NETWORK and EXT TARGETS Blocks. 574
4.7 Time Series Catalog 583
4.7(1) Catalog for PERLND Module 584
4.7(2) Catalog for IMPLND Module 602
4.7(3) Catalog for RCHRES Module 608
4.7(11) Catalog for COPY Module 623
4.7(12) Catalog for PLTGEN Module 624
4.7(13) Catalog for the DISPLY Module. 625
4.7(14) Catalog for the DURANL Module. 626
4.7(15) Catalog for the GENER Module 627
4.8 FORMATS Block. 629
4.9 Sequential File Formats 630
4.10 SPEC-ACTIONS Block 632

5.0 Sample RUN Data Sets 634

FIGURES

Number Page

4.7(1)-1 Groups of time series associated with the PERLND module . . 585
4.7(2)-1 Groups of time series associated with the IMPLND module . . 603
4.7(3)-1 Groups of time series associated with the RCHRES module . . 609
4.7(11)-1 Groups of time series associated with the COPY module . . . 628
4.7(12)-1 Groups of time series associated with the PLTGEN module . . 628

1.0 GENERAL INFORMATION AND CONVENTIONS

1.1 The User's Control Input

The User's Control Input (UCI) consists of a number of text lines, 80 characters wide in card images. A general feature of the UCI is that the card images are collected into groups. Groups may contain subordinate groups; that is, they may be nested. In every case, a group commences with a heading (such as, RUN) and ends with a delimiter (such as, END RUN).

The HSPF system will ignore any line in the UCI which contains three or more consecutive asterisks (***), just as a Fortran compiler bypasses comments in a source program. Blank lines are also ignored. This feature can be used to insert headings and comments which make the text more intelligible to the reader, but are not required by the HSPF system itself.

The body of the User's Control Input consists of one or more major groups of text, called data sets:

```
<data set 1>
<data set 2>
  -----
  -----
```

A data set is either a TSSMGR data set or a RUN data set. A TSSMGR data set consists of one or more commands which direct the time series store manager module to create, modify, or destroy labels of individual data sets in the TSS. A RUN data set contains all the data needed to perform a single RUN. A RUN is a set of operations with a common START date-time and END date-time.

1.2 General Comments on Method of Documentation

The documentation of each portion of the UCI is divided into three sections: "layout", "details", and "explanation".

The "layout" section shows how the input is arranged. Text always appearing in the same form (e.g. TSSM) is shown in upper case. Text which varies from job to job is shown by lower case symbols enclosed in angle brackets (<spa>). Lines containing illustrative text, not actually required by the system, have three consecutive asterisks, just as they might have in the UCI. Optional material, or that which is not always required, is enclosed in brackets []. The column numbers printed at the head of each layout show the exact starting location of each keyword and symbol.

The "details" section describes the input values required for each symbol appearing in the layout. The Fortran identifiers used to store the value(s) are given, followed by the format. The field(s) specified in this format start in the column containing the < which immediately precedes the symbol in the layout.

For example, < ds> in a TSSM data set starts in col 26 and ends in column 26 + 5 - 1 = column 30. Where relevant, the Details section also indicates default values and min and max values for each item in the UCI.

The "explanation" section contains any necessary explanatory material which could not fit into the details section.

2.0 FORMAT OF A TSSMGR DATA SET

A TSSMGR Data Set starts with a TSSM heading and ends with an END TSSM delimiter. The data set contains one or more commands and associated data, which may appear in any sequence. A single exception applies: DATASET NO, if required, must appear as the first input item in its group. All variables, numbers or strings, must be right-justified and end in column 30, except the LOCATION string.

Note that a separate program, NEWTSS, must be run to create and initialize a Time Series Store before it may be used by the HSPF system. (This stand-alone program is documented in Appendix III.)

2.1 Layout

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************

TSSM               TSSFL=< ts>

  ADD
    ***  creates a label for a new dataset in the TSS
    ***  the following group is repeated for each dataset:
    DATASET NO=          < ds>
      SPACE=             <spa>
      NAME=             <name>
      TIMESTEP=         <tsec>
      NMEMS=             <  m>
      ***  optional parameters:
     [STATION=]         <stat>
     [SECURITY=]         <sec>
     [UNITS=]          <units>
     [COMPRESSION=]     <comp>
     [OBS TIME=]         <  o>
     [FILLER CODE=]      < pv>
     [GAP CODE=]            gc
     [YEAROR=]             <y>
     [BASEYR=]           <byr>
```

```
[LOCATION=] <----------location----------------->
 ***  the following group is repeated for each member
 ***  (required at least once):
 MEMBER NAME=      <  mn>
  [NCOMPS=]         <  n>

     [KIND=]           <knd>
     [FORMAT=]         <  f>

UPDATE
  *** updates selected fields in the label of a dataset
  *** the following group is repeated for each dataset:
  DATASET NO=          < ds>
  *** any or all of the following is entered:
   [SECURITY=]         <sec>
   [OBS TIME=]         <  o>
   [UNITS=]          <units>
   [STATION=]         <stat>
   [YEAROR=]             <y>
   [BASEYR=]           <byr>
   [LOCATION=]  <       location                     >
    *** optional group, is repeated for each member:
   [MEMBER NAME=]     <  mn>
     [FORMAT=]         <  f>

SCRATCH
  *** deletes a dataset label (and,effectively, the dataset contents)
  *** the following line is repeated for each dataset:
  DATASET NO=          < ds>

EXTEND
  *** allocates more space to a dataset or to the TSS directory
  *** the following group is repeated for each dataset:
  DATASET NO=          < ds>
    *** allocate more space to a dataset:
    SPACE=             <spa>
    *** or to extend the directory:
  DIRECTORY   (not yet implemented)
    MAXDS=             < nd>

COMPACT       (not yet implemented)

SHOWSPACE
  *** show the free space in the TSS

SHOWDSL
  *** display the contents of the label of one or all the datasets:
  *** the following card is repeated for each dataset.  If omitted,
  *** all dataset labels will be displayed.
 [DATASET NO=]         < ds>
```

```
  SHOWTSS
    *** display the current state of the TSS

END TSSM

*******************************************************************************
```

2.2 Details

Symbol	Fortran Name(s)	Format	Comment
<ts>	TSSFL	I5	The Fortran unit no. of the TSS. Default 15
< nd>	TOTDS	I5	The total number of datasets.
< ds>	DSNO	I5	A unique identifying number for a dataset.
<spa>	SPACE	I5	The space reserved for a dataset in records. See Appendix III for formula.
<sec>	SECURE	A5	The read/write security for a dataset. READ, WRITE can be entered. Default: WRITE
<name>	NAME	A6	Name assigned to a dataset.
< f>	FMT	I5	The number of decimal digits desired in the output format. Default 0.
<units>	UNITS	A7	System of units used for the data stored in the TSS. Valid values are: ENGLISH or METRIC. Default METRIC.
<comp>	DSCMPR	A6	Compression indicator. Valid values are UNCOMPressed and COMPRessed Default: UNCOMPressed
< o>	OBSTIM	I5	Observation hour for daily data. Default is 24.
<stat>	STA	A6	Station identifier for a dataset.
< pv>	FILLVAL	A5	The padding value used to fill in missing data. This can be ZERO or UNDEFined. Default is UNDEFined.

gc	GAP	A2	Compression indicator for filled values preceding and following period of valid input within the year. May be UU, UC, CU or CC. Default UU. See note below.
<tsec>	DSDELT	I6	The time step in minutes for a dataset. The minimum value allowed is 1 minute.
< m>	MEMBERS	I5	Number of members in this dataset. (maximum=20)
<y>	YEAROR	A3	YES means a file is in yearly chronological order; otherwise, NO. Default NO.
<byr>	BASEYR+1	I5	The first year for which data can be stored. Default 1900.
<-location->	LOCATN	A65	Location description.
< mn>	MEMBER	A6	Name of the member, e.g. RAIN,EVAP.
< n>	MCOMP	I5	The number of components in a member. If RAIN(1), RAIN(2), RAIN(3) are components, then MCOMP=3. Each member has at least 1 component. Default 1.
<knd>	MKIND	A5	The kind of data in this member, POINT or MEAN. Default MEAN.

2.3 Explanation

Each input item must be right justified within its field. For example, TOTDS is input with I5 format; a value of 128 is input as bb128.

A data set consists of one or more members. The members may be further subdivided into components. Each component contains a single time series (eg. PREC 1). The maximum number of members and their components is 20; thus, a data set may hold up to 20 time series.

The parameter GAP was included to permit some compression of space, even where data are stored in uncompressed form. If the first letter of GAP is C, and data which start part-way through a calendar year are fed into the TSS dataset, the period prior to the start of data will be compressed. Note that this implies that data for the compressed period cannot subsequently be read in.

Similarly, if the second letter of GAP is C, and data which end part-way through a calendar year are fed into the TSS dataset, the period after the end of data will be compressed. (Note that this period could subsequently be filled with data, using the ADD or REPL access mode, provided space is available in the dataset).

To illustrate the above, consider the following example:
Suppose we need to store uncompressed data with a timestep of 1 minute for one month (say July 1974). According to the formula in Appendix III, a full calendar year would require 1041 records. But, if GAP=CC were used to compress the months Jan through June and August through December, the space required is only 88 records.

3.0 SAMPLE TSSMGR DATA SET

A sample input stream which creates the label for a dataset with only one member/component follows:

```
TSSM

  ADD
    DATASET NO=             39
    SPACE=                 100
    SECURITY=             READ
    NAME=               PRECIP
    UNITS=              METRIC
    COMPRESSION=         COMPR
    STATION=            US3112
    FILLER CODE=          ZERO
    TIMESTEP=              360
    NMEMS=                   1
    YEAROR=                YES
    LOCATION=   PALO ALTO, CALIFORNIA
    MEMBER NAME=          RAIN
      NCOMPS=                1
      KIND=               MEAN

END TSSM
```

Note: The steps required to create a TSS are explained in Appendix III.

4.0 FORMAT OF A RUN DATA SET

4.1 Summary

A RUN data set starts with a RUN heading and ends with an END RUN delimiter. The body of the text consists of several groups, called "blocks," which may appear in any sequence:

RUN

GLOBAL Block

Contains information of a global nature. It applies to every operation in the RUN.

OPN SEQUENCE Block

Specifies the operations to be performed in the RUN, in the sequence they will be executed. It indicates any grouping (INGROUPs).

<Operation-type> Block

Deals with data "domestic" to all the operations of the same <Operation-type>, for example, parameters and inital conditions for all Pervious Land-segments in a RUN. It is not concerned with relationships between operations, or with external sources or targets for time series. There is one <Operation-type> Block for each <operation-type> involved in the RUN.

[FTABLES Block]

A collection of function tables (FTABLES). A function table is used to document, in discrete numerical form, a functional relationship between two or more variables.

EXT SOURCES Block

Specifies time series which are input to the operations from external sources (TSS or sequential files).

[NETWORK Block]

Specifies any time series which are passed between operations.

[EXT TARGETS Block]

Specifies those time series which are output from operations to external destinations (TSS, sequential files).

FORMATS Block

Contains any user-supplied formats which may be required to read or record time series on external sequential files.

END RUN

Usually, a RUN data set will not include all of the above blocks. Their presence will be dictated by the operations performed in the RUN and the options which are selected.

4.2 GLOBAL Block

This block must always be present in a RUN data set.

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

GLOBAL
  <------------------------------- run-info ----------------------------------->
  START   <---s-date-time----> END<---e-date-time---->
  RUN INTERP OUTPUT LEVEL<lev>
  RESUME <res> RUN <run> TSSFL <tss>
END GLOBAL

********************************************************************************
Example
*******

GLOBAL
  Simulation of the most complex network ever attempted by mankind
  START       1970/01/01 00:00  END    1977/12/31 12:00
  RUN INTERP OUTPUT LEVEL    7
  RESUME     1 RUN     1
END GLOBAL
********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<run-info>	RUNINF(20)	A78	none	none	none
<s-date-time>	SYR,	I8,	none	1	32767
	SMO,	1X,I2,	1	1	12
	SDA,	1X,I2,	1	1	varies
	SHR,	1X,I2,	0	0	23
	SMI	1X,I2	0	0	59
<e-date-time>	EYR,	I8,	none	1	32767
		1X,I2,	12	1	12
	EDA,	1X,I2,	varies	1	varies
	EHR,	1X,I2,	#24	0	24 #only if EMI is 0
	EMI	1X,I2	0	0	59
<lev>	OUTLEV	I5	0	0	10
<res>	RESMFG	I5	0	0	1
<run>	RUNFG	I5	0	0	1
<tss>	TSSFL	I5	15	15	22

Explanation

RUNINF stores the users comments regarding the RUN.

Users conventionally label the same point in time differently, depending whether they are looking forwards or backwards towards it. For example, if we say that a RUN starts on 1978/05 we mean that it commences at the start of May 1978. On the other hand, if we say it ends on 1978/05 we mean it terminates at the end of May 1978. Thus, HSPF has two separate conventions for the external labeling of time. When supplying values for a date/time field a user may omit any element in the field except the year, which must be supplied (as a 4 digit figure). HSPF will substitute the defaults given above for any blank or zero values. The completed starting and ending date/time fields are translated into another format, which is the only one used to label intervals and time points internally. It has a resolution of 1 minute. Thus, time is recorded as a year/month/day/hour/minute set, to completely specify either a time interval or a point. The date/time used by the internal clock uses the "contained within" principle. For example, the first minute in an hour is numbered 1 (not 0) and the last is numbered 60 (not 59). The same applies to the numbering of hours. Thus, the time conventionally labeled 11:15 is in the 12th hour of the day so is labeled 12:15 internally; the last minute of 1978 is labeled 1978/12/31 24:60. This convention is extended to the labeling of points by labeling a point with the minute which immediately precedes it. Thus, midnight New Year's eve

1978/1979 is 1978/12/31 24:60, not 1979/01/01 00:00. This gives a system for uniquely labeling each point internally.

OUTLEV is a flag which governs the quantity of informative output produced by the Run Interpreter. A value 0 results in minimal output; 10 in the maximum. It does not affect error or warning messages.

If RESMFG is 1, the system will operate in "resume" mode; that is, it will use the same data as were supplied in a previous RUN data set except where overriding information is supplied in this data set. (This feature is not supported in the current release of HSPF).

If RUNFG is 1, the system will both interpret and execute the RUN. If it is 0, only interpretation will be done. This feature is useful if one wishes to debug an input stream during the day time, but defer execution to a night block.

TSSFL is the Fortran unit number of the Time Series Store. It defaults to 15. Note that some Fortran compilers (eg IBM 370) require a DEFINE FILE statement in which the file (TSS) length is specified, in records. For this purpose, HSPF has several DEFINE FILE statements, each of which refers to a TSS of different length. Select the length of your TSS and supply a value for TSSFL which refers to the appropriate DEFINE FILE statement.

4.3 OPN SEQUENCE Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
OPN SEQUENCE
   [INGRP              INDELT <idt>]
      <-opn-id----->
      . . . . . . .
      . . . . . . .
      <-opn-id----->
   [END INGRP  ]
      <-opn-id----->   INDELT <idt>
      . . . . . . .
      . . . . . . .
      <-opn-id----->   INDELT <idt>
   [INGRP              INDELT <idt>]
      <-opn-id----->
      . . . . . . .
      . . . . . . .
   [END INGRP  ]
    . . . . . .
    . . . . . .
END OPN SEQUENCE
```

```
*******
Example
*******
OPN SEQUENCE
    INGRP                INDELT 02:00
      PERLND      20
      PERLND      21
      PERLND      22
    END INGRP
      RCHRES       1     INDELT 12:00
END OPN SEQUENCE
********************************************************************************
```

Details

Symbol	Fortran Name(s)	Format	Comment
<idt>	HRMIN(2)	I2,1X,I2	Time interval (hour:min) used in the INPAD e.g. 00:05
<-opn-id->	OPTYP,OPTNO	A6,5X,I3	Type and no. of this operation. e.g. RCHRES 100

Explanation

This block specifies the various operations to be performed in the RUN and, optionally, their grouping into INGROUPs. The operations will be performed in the sequence specified in the block, apart from repetition implied by grouping.

Every <-opn-id-> consists of OPTYP and OPTNO. The OPTYP field must contain an identifier of up to 6 characters which corresponds to an "operating module id" in the HSPF system. The OPTNO field contains an integer which distinguishes operations of the same type from one another. Every <opn-id> must be unique.

The time intervals of the scratch pads used in the RUN are specified in this block. These appear on the INGROUP lines, except where the user has not placed an operation in an INGROUP. In that case <idt> is specified alongside <-opn-id->.

4.4 <Operation-type> Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

<otyp>
  General input
  Section 1 input  -
  Section 2 input   |  Only supplied if the operating module is sectioned
       .            |  and the section is active
       .            |
  Section N input  -
END <otyp>

********************************************************************************
```

Details

Symbol	Fortran Name(s)	Format	Comment
<otyp>	OPTYP	A6	Type of operation covered in this block, e.g. RCHRES, PERLND

Explanation

This type of block deals with data which are "domestic" to all operations of the same <Operation-type>, e.g. the parms and initial conds for all the Pervious Land- segments in a RUN. It is not concerned with relationships between operations or with external sources or targets for time series.

This type of block provides for "general" input and for input which is specific to individual "sections" of the OM. The latter only apply to modules which are sectioned. The general input contains all of the information which simple (non-sectioned) modules require; for sectioned modules it contains input which is not specific to any one section.

The general organization of the <Operation-type> blocks is as follows:

The user supplies his input in a set of tables (eg. ACTIVITY, Sect 4.4(1).1.1 below). Each table has a name (eg. ACTIVITY), called the "Table-type". A table starts with the heading <Table-type> and ends with the delimiter END <Table-type>. The body of the table consists of:

<range><------------values-------------------------->

where <range> is the range of operation-type numbers to which the <values> apply. If the second field in <range> is blank, the range is assumed to consist of a single operation. Thus, in the example in Sect 4.4(1).1.1, Pervious Land-segments (PLSs) 1 through 7 have the same set of active sections, while segment 9 has a different set.

Thus, a table lists the values given to a specified set of variables (occupying only 1 line) for all the operations of a given type. The input was designed this way to minimize the quantity of data supplied when many operations have the same values for certain sets of input.

HSPF will only look for a given Table-type if the options already specified by the user require data contained within it. Thus, Table-type MON-INTERCEP (Sect 4.4(1).4.5) is relevant only if VCSFG in Table-type PWAT-PARM1 (4.4(1).4.1) is set to 1 for one or more PLSs. The system has been designed to ignore redundant information. Thus, if VCSFG is 0 and Table-type MON-INTERCEP is supplied, the table will be ignored.

On the other hand, if an expected value is not supplied, the system will attempt to use a default value. This situation can arise in one of three ways:

1. The entire table may be missing from the UCI.

2. The table may be present but not contain an entry (line) for the operation in question. The example in Sect 4.4(1).1.1 has no entry for PLS No. 8. Thus, all values in its active sections vector will acquire the default of 0.

3. A field may be left blank or given the value zero. In the example in Sect 4.4(1).4.2 , KVARY will acquire the default value 0.0, for PLSs 1 through 7.

When appropriate, the HSPF system will also check that a value supplied by the user falls within an allowable range. If it does not, an error message is generated.

Note that a table contains either integers or real values, but not both. For example, Table-type ACTIVITY (Sect 4.4(1).1.1) contains only integer flags, but Table-type PWAT-PARM2 (4.4(1).4.2) contains only real numbers. For tables containing real-valued data, the documentation gives separate defaults, minima and maxima for the English and Metric unit systems. The user specifies the system in which he is working, in Table-type GEN-INFO (eg. Sect 4.4(1).1.3).

4.4(1) PERLND Block

```
********************************************************************************
              1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

PERLND
  General input
 [section ATEMP input]
 [section SNOW input]
 [section PWATER input]
 [section SEDMNT input]
 [section PSTEMP input]
 [section PWTGAS input]
 [section PQUAL input]
 [section MSTLAY input]
 [section PEST input]
 [section NITR input]
 [section PHOS input]
 [section TRACER input]
END PERLND

********************************************************************************
```

Explanation

This block contains the data which are "domestic" to all the Pervious Land-segments in the RUN. The "General input" is always relevant: other input is only required if the module section concerned is active.

4.4(1).1 PERLND BLOCK -- General input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  Table-type ACTIVITY
 [Table-type PRINT-INFO]
  Table-type GEN-INFO

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] above are not always required; for example, because all the values can be defaulted.

4.4(1).1.1 Table-type ACTIVITY -- Active Sections Vector

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ACTIVITY
  <-range><-----------------a-s-vector-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END ACTIVITY

*******
Example
*******

  ACTIVITY
    <PLS >               Active Sections                                 ***
    # -  # ATMP SNOW PWAT  SED  PST  PWG PQAL MSTL PEST NITR PHOS TRAC***
    1    7    1    1    1
    9         0    0    0    1
  END ACTIVITY

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<a-s-vector>	ASVEC(12)	12I5	0	0	1

Explanation

The PERLND module is divided into 12 sections. The values supplied in this table specify which sections are active and which are not, for each operation involving the PERLND module. A value of 0 means "inactive" and 1 means "active". Any meaningful subset of sections may be active.

4.4(1).1.2 Table-type PRINT-INFO -- Printout information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PRINT-INFO
  <-range><-----------------------print-flags------------------------><piv><pyr>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PRINT-INFO

*******
Example
*******

  PRINT-INFO
    <PLS > ********************* Print-flags ************************* PIVL  PYR
    # -  # ATMP SNOW PWAT  SED  PST  PWG PQAL MSTL PEST NITR PHOS TRAC *********
    1    7    2    4    6                                  4    3    2   10   12
  END PRINT-INFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<print-flags>	PFLAG(12)	12I5	4	2	6
<piv>	PIVL	I5	1	1	1440
<pyr>	PYREND	I5	9	1	12

Explanation

HSPF permits the user to vary the printout level (maximum frequency) for the various active sections of an operation. The meaning of each permissible value for PFLAG() is:

2 means every PIVL intervals
3 means every day
4 means every month
5 means every year
6 means never

In the example above, output from Pervious Land-segments 1 thru' 7 will occur as follows:

Section	Max frequency
ATEMP	10 intervals
SNOW	month
PWATER	never
SEDMNT	¦
thru'	¦ month (defaulted)
PEST	¦
NITR	month
PHOS	day
TRACER	10 intervals

A value need only be supplied for PIVL if one or more sections have a printout level of 2. For those sections, printout will occur every PIVL intervals (that is, every PDELT=PIVL*DELT mins). PIVL must be chosen such that there are an integer no. of PDELT periods in a day.

HSPF will automatically provide printed output at all standard intervals greater than the specified minimum interval. In the above example, output for section PHOS will be printed at the end of each day, month and year.

PYREND is the calendar month which will terminate the year for printout purposes. Thus, the annual summary can reflect the situation over the past water year or the past calendar year, etc.

4.4(1).1.3 Table-type GEN-INFO — Other general information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  GEN-INFO
  <-range><---PLS-id---------><blk><--unit-syst--><-printu->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END GEN-INFO

*******
Example
*******

  GEN-INFO
    <PLS >      Name          NBLKS   Unit-systems   Printer***
    # -  #                          User  t-series Engl Metr***
                                           in  out        ***
    1      Yosemite Valley
    2      Kings river            2    1    1        23
  END GEN-INFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<PLS-id>	LSID(10)	10A2	none	none	none
<blk>	NBLKS	I5	1	1	5
<unit-syst>	UUNITS,IUNITS, OUNITS	3I5	1	1	2
<printu>	PUNIT(2)	2I5	0	6	99

Explanation

Any string of up to 20 characters may be supplied as the identifier for a PLS.

NBLKS is the no. of "blocks" into which the surface and near-surface zones of the PLS will be subdivided for simulation purposes. It affects the PWATER, SEDMNT, and agricultural chemical sections of the PERLND module. See the functional description of those sections for further details.

The values supplied for <unit-syst> indicate the system of units for data in the UCI, input time series and output time series respectively: 1 means English units, 2 means Metric units.

The values supplied for <printu> indicate the destinations of printout in English and Metric units respectively. A value 0 means no printout is required in that system. A non-zero value means printout is required in that system and and the value is the Fortran unit no. of the file to which the printout is to be written. Note that printout for each Pervious Land Segment can be obtained in either the English or Metric systems, or both (irrespective of the system used to supply the inputs).

4.4(1).2 PERLND BLOCK -- Section ATEMP input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type ATEMP-DAT]

********************************************************************************
```

Explanation

The exact format of the table mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] above are not always required; for example, because all the values can be defaulted.

4.4(1).2.1 Table-type ATEMP-DAT -- Elevation difference between gage & PLS

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ATEMP-DAT
  <-range><el-diff-><-airtmp->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END ATEMP-DAT

*******
Example
*******

  ATEMP-DAT
    <PLS >   El-diff   ***
    # - #     (ft)     ***
    1    7    150.
  END ATEMP-DAT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<el-diff>	ELDAT	F10.0	0.0	none	none	ft	Engl
			0.0	none	none	m	Metric
<airtmp>	AIRTMP	F10.0	60	-60	140	Deg F	Engl
			15	-50	60	Deg C	Metric

Explanation

ELDAT is the difference in elevation between the temp gage and the PLS; it is used to estimate the temp over the PLS by application of a lapse rate. It is positive if the PLS is higher than the gage, and vice versa.

AIRTMP is the air temperature over the PLS at the start of the RUN.

4.4(1).3 PERLND BLOCK -- Section SNOW input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type ICE-FLAG]
  Table-type SNOW-PARM1
 [Table-type SNOW-PARM2]
 [Table-type SNOW-INIT1]
 [Table-type SNOW-INIT2]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] above are not always required; for example, because all the values can be defaulted.

4.4(1).3.1 Table-type ICE-FLAG -- governs simulation of ice formation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ICE-FLAG
  <-range><ice>
  . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . .
  END ICE-FLAG

*******
Example
*******

  ICE-FLAG
    <PLS > Ice-   ***
    # -  # flag   ***
    1    7    1
  END ICE-FLAG

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<ice>	ICEFG	I5	0	0	1

Explanation

A value 0 means ice formation in the snow pack will not be simulated; 1 means it will.

4.4(1).3.2 Table-type SNOW-PARM1 -- First group of SNOW parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SNOW-PARM1
  <-range><----------------snowparm1------------------------>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END SNOW-PARM1

*******
Example
*******

  SNOW-PARM1
    <PLS >  Latitude     Mean-     SHADE    SNOWCF    COVIND***
    # -  #                elev                              ***
    1    7    39.5       3900.       0.3       1.2        10
  END SNOW-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<snowparm1>	LAT,	5F10.0	40.0	-90.0	90.0	degrees	Both
	MELEV,		0.0 0.0	0.0 0.0	30000.0 10000.0	ft m	Engl Metric
	SHADE,		0.0	0.0	1.0	none	Both
	SNOWCF,		none	1.0	100.0	none	Both
	COVIND		none none	0.01 0.25	none none	in mm	Engl Metric

Explanation

LAT is the latitude of the PLS. It is positive for the northern hemisphere, negative for the southern hemisphere.

MELEV is the mean elevation of the PLS.

SHADE is the fraction of the PLS which is shaded from solar radiation by, for example, trees.

SNOWCF is the factor by which recorded precip data will be multiplied if the simulation indicates it is snowfall, to account for poor catch efficiency under snow conditions.

COVIND is the maximum pack (water equivalent) at which the entire PLS will be covered with snow (see functional description of SNOW section).

4.4(1).3.3 Table-type SNOW-PARM2 -- Second group of SNOW parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SNOW-PARM2
  <-range><---------------------snowparm2----------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      SNOW-PARM2

*******
Example
*******

  SNOW-PARM2
    <PLS >                                                            ***
    # -  #     RDCSN     TSNOW    SNOEVP    CCFACT    MWATER    MGMELT***
    1    7       0.2       33.
  END SNOW-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<snowparm2>	RDCSN,	F10.0	0.15	0.01	1.0	none	Both
	TSNOW,	F10.0	32.0	30.0	40.0	degF	Engl
			0.0	-1.0	5.0	degC	Metric
	SNOEVP,	F10.0	0.1	0.0	1.0	none	Both
	CCFACT,	F10.0	1.0	0.0	2.0	none	Both
	MWATER,	F10.0	0.03	0.0	1.0	none	Both
	MGMELT	F10.0	0.01	0.0	1.0	in/day	Engl
			0.25	0.0	25.	mm/day	Metric

Explanation

RDCSN is the density of cold, new snow relative to water. This value applies to snow falling at air temps <= OdegF. At higher temperatures the density of snow is adjusted.

TSNOW is the air temp below which precip will be snow, under saturated conditions. Under non-saturated conditions the temperature is adjusted slightly.

SNOEVP is a parameter which adapts the snow evaporation (sublimation) equation to field conditions.

CCFACT is a parameter which adapts the snow condensation/convection melt equation to field conditions.

MWATER is the max water content of the snow pack, in depth water per depth water equiv.

MGMELT is the max rate of snowmelt by ground heat, in depth of water equiv per day. This is the value which applies when the pack temperature is at freezing point.

4.4(1).3.4 Table-type SNOW-INIT1 -- First group of initial values

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SNOW-INIT1
  <-range><-------------------snowinit1------------------------------>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END SNOW-INIT1

*******
Example
*******

  SNOW-INIT1
    <PLS >                                                            ***
    # -  # Pack-snow  Pack-ice Pack-watr    RDENPF      DULL    PAKTMP***
    1    7       2.1                .02       .40
  END SNOW-INIT1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<snowinit1>	Pack-snow	F10.0	0.0	0.0	none	in	Engl
			0.0	0.0	none	mm	Metric
	Pack-ice	F10.0	0.0	0.0	none	in	Engl
			0.0	0.0	none	mm	Metric
	Pack-watr	F10.0	0.0	0.0	none	in	Engl
			0.0	0.0	none	mm	Metric
	RDENPF	F10.0	0.2	.01	1.0	none	Both
	DULL	F10.0	400.	0.0	800.	none	Both
	PAKTMP	F10.0	32.	none	32.	degF	Engl
			0.0	none	0.0	degC	Metric

Explanation

Pack-snow is the quantity of snow in the pack (water equiv).

Pack-ice is the quantity of ice in the pack (water equiv)

Pack-watr is the quantity of liquid water in the pack.

RDENPF is the density of the frozen contents (snow+ice) of the pack, relative to water.

DULL is an index to the dullness of the pack surface, from which albedo is estimated.

PAKTMP is the mean temperature of the frozen contents of the pack.

4.4(1).3.5 Table-type SNOW-INIT2 — Second group of initial values

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SNOW-INIT2
  <-range><--------snowinit2----------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END SNOW-INIT2

*******
Example
*******

  SNOW-INIT2
    <PLS >                        ***
    # - #   COVINX  XLNMLT  SKYCLR***
    1   7                    0.50
  END SNOW-INIT2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<snowinit2>	COVINX,	F10.0	0.01	0.01	none	in	Engl
			0.25	0.25	none	mm	Metric
	XLNMLT,	F10.0	0.0	0.0	none	in	Engl
			0.0	0.0	none	mm	Metric
	SKYCLR	F10.0	1.0	.15	1.0	none	Both

Explanation

COVINX is the current pack (water equiv) required to obtain complete areal coverage of the PLS. If the pack is less than this amount, areal cover is prorated (PACKF/COVINX).

XLNMLT is the current remaining possible increment to ice storage in the pack (see functional description). This value is only relevant if ice formation is being simulated (ICEFG= 1).

SKYCLR is the fraction of sky which is assumed to be clear at the present time.

In the above example COVINX and XLNMLT will be assigned default values because the user has left the fields blank.

4.4(1).4 PERLND BLOCK -- Section PWATER input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type PWAT-PARM1  ]
  Table-type PWAT-PARM2
 [Table-type PWAT-PARM3  ]
  Table-type PWAT-PARM4
 [Table-type MON-INTERCEP]   -
 [Table-type MON-UZSN    ]   |
 [Table-type MON-MANNING ]   | only reqd if the relevant quantity
 [Table-type MON-INTERFLW]   | varies through the year
 [Table-type MON-IRC     ]   |
 [Table-type MON-LZETPARM]   -
 [Table-type PWAT-STATE1 ]
 [Table-type PWAT-BLKSTAT]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] above are not always required; for example, because all the values can be defaulted.

4.4(1).4.1 Table-type PWAT-PARM1 -- First group of PWATER parms (flags)

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWAT-PARM1
  <-range><--------------pwatparm1------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PWAT-PARM1

*******
Example
*******

  PWAT-PARM1
    <PLS >                   Flags                          ***
    # -  # CSNO RTOP UZFG  VCS  VUZ  VNN VIFW VIRC  VLE     ***
    1    7    1    1
  END PWAT-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<pwatparm1>	CSNOFG,	I5	0	0	1
	RTOPFG,	I5	0	0	1
	UZFG,	I5	0	0	1
	VCSFG,	I5	0	0	1
	VUZFG,	I5	0	0	1
	VNNFG,	I5	0	0	1
	VIFWFG,	I5	0	0	1
	VIRCFG,	I5	0	0	1
	VLEFG	I5	0	0	1

Explanation

If CSNOFG is 1, section PWATER assumes that snow accumulation and melt is being considered. It will, therefore, expect that the time series produced by section SNOW are available, either internally (produced in this RUN) or from external sources (produced in a previous RUN). If CSNOFG is 0, no such time series are expected. See the functional description for further information.

If RTOPFG is 1, routing of overland flow is done in exactly the same way as in HSPX, ARM and NPS. A value of 0 results in a new algorithm being used.

If UZFG is 1, inflow to the upper zone is computed in the same way as in HSPX, ARM and NPS. A value of zero results in the use of a new algorithm, which should be less sensitive to changes in DELT.

The flags beginning with "V" indicate whether or not certain parameters will be assumed to vary through the year: 1 means they do vary, 0 means they do not. The quantities concerned are:

VCSFG	interception storage capacity
VUZFG	upper zone nominal storage
VNNFG	Manning's n for the overland flow plane
VIFWFG	interflow inflow parameter
VIRCFG	interflow recession const
VLEFG	lower zone E-T parameter

If any of these flags are on, monthly values for the parameter concerned must be supplied (see Table-types MON- , documented later).

4.4(1).4.2 Table-type PWAT-PARM2 -- Second group of PWATER parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWAT-PARM2
  <-range><--------------------------pwatparm2---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PWAT-PARM2

*******
Example
*******

  PWAT-PARM2
    <PLS > ***
    #  - # ***FOREST      LZSN    INFILT      LSUR     SLSUR     KVARY     AGWRC
     1   7       0.2       8.0       0.7      400.      .001                 .98
  END PWAT-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwatparm2>	FOREST,	F10.0	0.0	0.0	1.0	none	Both
	LZSN,	F10.0	none	.01	100.	in	Engl
			none	.25	2500.	mm	Metric
	INFILT,	F10.0	none	0.0001	100.	in/hr	Engl
			none	0.0025	2500.	mm/hr	Metric
	LSUR,	F10.0	none	1.0	none	ft	Engl
			none	0.3	none	m	Metric
	SLSUR,	F10.0	none	.000001	10.	none	Both
	KVARY,	F10.0	0.0	0.0	none	1/in	Engl
			0.0	0.0	none	1/mm	Metric
	AGWRC	F10.0	none	0.001	1.0	1/day	Both

Explanation

FOREST is the fraction of the PLS which is covered by forest which will continue to transpire in winter.

LZSN is the lower zone nominal storage.

INFILT is an index to the infiltration capacity of the soil.

LSUR is the length of the assumed overland flow plane, and SLSUR is the slope.

KVARY is a parameter which affects the behavior of groundwater recession flow, enabling it to be non exponential in its decay with time.

AGWRC is the basic groundwater recession rate if KVARY is zero and there is no inflow to groundwater (rate of flow today/ rate yesterday).

In the above example, KVARY will be assigned the default value of 0.0 .

4.4(1).4.3 Table-type PWAT-PARM3 — Third group of PWATER parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWAT-PARM3
  <-range><------------------------pwatparm3----------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PWAT-PARM3

*******
Example
*******
  PWAT-PARM3
    <PLS >***
    # -  #*** PETMAX    PETMIN    INFEXP    INFILD    DEEPFR    BASETP    AGWETP
    1    7
    9             39        33       3.0       1.5
  END PWAT-PARM3

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwatparm3>	PETMAX,	F10.0	40. 4.5	none none	none none	degF degC	Engl Metric
	PETMIN,	F10.0	35. 1.7	none none	none none	degF degC	Engl Metric
	INFEXP,	F10.0	2.0	0.0	10.0	none	Both
	INFILD,	F10.0	2.0	1.0	2.0	none	Both
	DEEPFR,	F10.0	0.0	0.0	1.0	none	Both
	BASETP,	F10.0	0.0	0.0	1.0	none	Both
	AGWETP	F10.0	0.0	0.0	1.0	none	Both

Explanation

PETMAX is the air temp below which E-T will arbitrarily be reduced below the value obtained from the input time series, and PETMIN is the temp below which E-T will be zero regardless of the value in the input time series. These values are only used if snow is being considered (CSNOFG= 1).

INFEXP is the exponent in the infiltration equation, and INFILD is the ratio between the max and mean infiltration capacities over the PLS.

DEEPFR is the fraction of groundwater inflow which will enter deep (inactive) groundwater and, thus, be lost from the system as it is defined in HSPF.

BASETP is the fraction of remaining potential E-T which can be satisfied from baseflow (groundwater outflow), if enough is available.

AGWETP is the fraction of remaining potential E-T which can be satisfied from active groundwater storage if enough is available.

In the above example, all parameters will be suppplied default values for Land-segments 1 through 7, while DEEPFR thru' AGWETP will be supplied defaults for Land-segment 9.

4.4(1).4.4 Table-type PWAT-PARM4 -- Fourth group of PWATER parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWAT-PARM4
  <-range><--------------------pwatparm4----------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PWAT-PARM4

*******
Example
*******

  PWAT-PARM4
    <PLS >                                                              ***
   # -   #     CEPSC      UZSN      NSUR     INTFW       IRC     LZETP***
    1    7       0.1       1.3       0.1        3.       0.5       0.7
  END PWAT-PARM4

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwatparm4>	CEPSC,	F10.0	0.0	0.0	10.0	in	Engl
			0.0	0.0	250.	mm	Metric
	UZSN,	F10.0	none	0.01	10.0	in	Engl
			none	0.25	250.	mm	Metric
	NSUR,	F10.0	0.1	0.001	1.0		Both
	INTFW,	F10.0	none	0.0	none	none	Both
	IRC,	F10.0	none	0.01	1.0	1/day	Both
	LZETP	F10.0	0.0	0.0	1.0	none	Both

Explanation

Values need only be supplied for those parameters which do not vary through the year. If they do vary (as specified in Table-type PWAT-PARM1), monthly values are supplied in the tables documented immediately below this one.

CEPSC is the interception storage capacity.

UZSN is the upper zone nominal storage.

NSUR is Manning's n for the assumed overland flow plane.

INTFW is the interflow inflow parameter.

IRC is the interflow recession parm. Under zero inflow, this is the ratio of interflow outflow rate today/ rate yesterday

LZETP is the lower zone E-T parm. It is an index to the density of deep-rooted vegetation.

4.4(1).4.5 Table-type MON-INTERCEP -- Monthly interception storage capacity

```
********************************************************************************
            1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-INTERCEP
  <-range><------------------mon-icep-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-INTERCEP

*******
Example
*******

  MON-INTERCEP
    <PLS >  Interception storage capacity at start of each month       ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .02  .03  .03  .04  .05  .08  .12  .15  .12  .05  .03  .01
  END MON-INTERCEP

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-icep>	CEPSCM(12)	12F5.0	0.0	0.0	10.	in	Engl
			0.0	0.0	250.	mm	Metric

Explanation

Only required if VCSFG in Table-type PWAT-PARM1 is 1.

4.4(1).4.6 Table-type MON-UZSN -- Monthly upper zone storage

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-UZSN
  <-range><-----------------mon-uzsn------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-UZSN

*******
Example
*******

  MON-UZSN
    <PLS >  Upper zone storage at start of each month                  ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC ***
    1    7  .30  .35  .30  .45  .56  .57  .45  .67  .64  .54  .56  .40
  END MON-UZSN

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-uzsn>	UZSNM(12)	12F5.0	none	.01	10.	in	Engl
			none	.25	250.	mm	Metric

Explanation

This table is only required if VUZFG in Table-type PWAT-PARM1 is 1.

4.4(1).4.7 Table-type MON-MANNING -- Monthly Manning's n values

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-MANNING
  <-range><---------------mon-Manning------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-MANNING

*******
Example
*******

  MON-MANNING
    <PLS >  Manning's n at start of each month                        ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC ***
    1    7  .23  .34  .34  .35  .28  .35  .37  .35  .28  .29  .30  .30
  END MON-MANNING

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-Manning>	NSURM(12)	12F5.0	.10	.001	1.0	complex	Both

Explanation

This table is only required if VNNFG in Table-type PWAT-PARM1 is 1.

4.4(1).4.8 Table-type MON-INTERFLW -- monthly interflow inflow parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-INTERFLW
  <-range><----------------mon-interflw------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-INTERFLW

*******
Example
*******

  MON-INTERFLW
    <PLS >  Interflow inflow parameter for start of each month          ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC  ***
    1    7  2.0  3.3  3.6  3.8  4.2  5.6  5.6  7.6  7.5  5.6  4.6  3.4
  END MON-INTERFLW

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-interflw>	INTFWM(12)	12F5.0	none	0.0	none	none	Both

Explanation

This table is only required if VIFWFG in Table-type PWAT-PARM1 is 1.

4.4(1).4.9 Table-type MON-IRC -- Monthly interflow recession constants

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-IRC
  <-range><--------------mon-irc------------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-IRC

*******
Example
*******

  MON-IRC
    <PLS >  Interflow recession constant at start of each month       ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .35  .40  .40  .40  .40  .43  .45  .45  .50  .45  .45  .40
  END MON-IRC

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-irc>	IRCM(12)	12F5.0	none	.01	1.0	/day	Both

Explanation

This table is only required if VIRCFG in Table-type PWAT-PARM1 is 1.

4.4(1).4.10 Table-type MON-LZETPARM -- Monthly lower zone E-T parameter

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-LZETPARM
  <-range><---------------mon-lzetparm------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-LZETPARM

*******
Example
*******

  MON-LZETPARM
    <PLS >  Lower zone evapotransp  parm at start of each month       ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .30  .30  .35  .35  .40  .40  .45  .45  .45  .45  .42  .38
  END MON-LZETPARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-lzetparm>	LZETPM(12)	12F5.0	0.0	0.0	1.0	none	Both

Explanation

This table is only required if VLEFG in Table-type PWAT-PARM1 is 1.

4.4(1).4.11 Table-type PWAT-STATE1 -- PWATER state variables

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWAT-STATE1
  <-range><-------------pwat-state1------------------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PWAT-STATE1

*******
Example
*******

  PWAT-STATE1
    <PLS >  PWATER state variables***
    # -  #***    CEPS      SURS       UZS      IFWS       LZS      AGWS      GWVS
    1    7       0.05      0.10      0.25      0.01       8.2       2.0      .025
  END PWAT-STATE1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwat-state1>	CEPS	7F10.0	.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	SURS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	UZS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	IFWS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	LZS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	AGWS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	GWVS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric

Explanation

This table is used to specify the initial water storages.

CEPS is the interception storage.
SURS is the surface (overland flow) storage.
UZS is the upper zone storage.
IFWS is the interflow storage.
LZS is the lower zone storage.
AGWS is the active groundwater storage.
GWVS is the index to groundwater slope; it is a measure of antecedent active groundwater inflow.

If NBLKS > 1, values in SURS, UZS, and IFWS fields are not processed, because these quantities are computed by averaging values in several PWAT-BLKSTAT tables.

4.4(1).4.12 Table-type PWAT-BLKSTAT -- Block-specific storages

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWAT-BLKSTAT
  <-range><--------pwat-blkstat-------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END PWAT-BLKSTAT

*******
Example
*******

  PWAT-BLKSTAT
    <PLS >  Storages in Block 2***
    # -  #     SURSB      UZSB     IFWSB***
    1    7      0.02      0.22      0.01
  END PWAT-BLKSTAT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwat-blkstat>	SURSB	3F10.0	0.001	0.001	100	inches	Engl
			0.025	0.025	2500	mm	Metric
	UZSB		0.001	0.001	100	inches	Engl
			0.025	0.025	2500	mm	Metric
	IFWSB		0.001	0.001	100	inches	Engl
			0.025	0.025	2500	mm	Metric

Explanation

If a PLS is subdivided into blocks (NBLKS > 1), certain initial storages are specified individually for each of the blocks. This table is repeated for each block.

SURSB is the surface storage.
UZSB is the upper zone storage.
IFWSB is the interflow storage.

4.4(1).5 PERLND BLOCK -- Section SEDMNT input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type SED-PARM1]            Tables in brackets [] are
  Table-type SED-PARM2             not always required.
  Table-type SED-PARM3
 [Table-type MON-COVER]
 [Table-type MON-NVSI]
 [Table-type SED-STOR]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

4.4(1).5.1 Table-type SED-PARM1 -- First group of SEDMNT parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SED-PARM1
  <-range><--sed-parm1-->
  . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . .
  END SED-PARM1

*******
Example
*******

  SED-PARM1
    <PLS >***
    # -  #  CRV VSIV SDOP***
    1    7    0    1    0
  END SED-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<sed-parm1>	CRVFG	3I5	0	0	1
	VSIVFG		0	0	1
	SDOPFG		0	0	1

Explanation

If CRVFG is 1, erosion-related cover may vary throughout the year. Values are supplied in Table-type MON-COVER.

If VSIVFG is 1, the rate of net vertical sediment input may vary throughout the year. Values are supplied in Table-type MON-NVSI.

If SDOPFG is 1, removal of sediment from the land surface will be simulated with the algorithm used in the ARM and NPS models. If it is 0, the new algorithm will be used.

4.4(1).5.2 Table-type SED-PARM2 — Second group of SEDMNT parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SED-PARM2
  <-range><--------------sed-parm2---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END SED-PARM2

*******
Example
*******

  SED-PARM2
    <PLS >***
    # -  #      SMPF      KRER      JRER     AFFIX     COVER      NVSI***
    1    7       0.9      0.08      1.90      0.01       0.5    -0.100
  END SED-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sedparm2>	SMPF	6F10.0	1.0	0.001	1.0	none	Both
	KRER		0.0	0.0	none	complex	Both
	JRER		none	none	none	complex	Both
	AFFIX		0.0	0.0	1.0	/day	Both
	COVER		0.0	0.0	1.0	none	Both
	NVSI		0.0	none	none	lb /ac.day	Engl
			0.0	none	none	kg /ha.day	Metric

Explanation

SMPF is a "supporting management practice factor." It is used to simulate the reduction in erosion achieved by use of erosion control practices.

KRER is the coefficient in the soil detachment equation.
JRER is the exponent in the soil detachment equation.
AFFIX is the fraction by which detached sediment storage decreases each day, as a result of soil compaction.
COVER is the fraction of land surface which is shielded from erosion by rainfall (not considering snow cover, which can be handled by simulation).
NVSI is the rate at which sediment enters detached storage from the atmosphere. A negative value can be supplied (e.g., to simulate removal by human activity or wind).

If monthly values for COVER and NVSI are being supplied, values supplied for these variables in this table are not relevant.

4.4(1).5.3 Table-type SED-PARM3 -- Third group of SEDMNT parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SED-PARM3
  <-range><-------------sed-parm3----------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END SED-PARM3

*******
Example
*******

  SED-PARM3
    <PLS >***
    # -  #     KSER      JSER      KGER      JGER***
    1    7     0.08       1.7      0.06       1.4
  END SED-PARM3

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sedparm3>	KSER	4F10.0	0.0	0.0	none	complex	Both
	JSER		none	none	none	complex	Both
	KGER		0.0	0.0	none	complex	Both
	JGER		none	none	none	complex	Both

Explanation

KSER and JSER are the coefficient and exponent in the detached sediment washoff equation.
KGER and JGER are the coefficient and exponent in the matrix soil scour equation (simulates gully erosion, etc.).

4.4(1).5.4 Table-type MON-COVER -- Monthly erosion related cover values

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-COVER
  <-range><---------------mon-cover---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-COVER

*******
Example
*******

  MON-COVER
    <PLS >  Monthly values for erosion related cover ***
    # -  # JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7 0.0  .12  .12  .24  .24  .56  .67  .56  .34  .34  .23  .12
  END MON-COVER

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-cover>	COVERM(12)	12F5.0	0.0	0.0	1.0	none	Both

Explanation

This table is only required if CRVFG in Table-type SED-PARM1 is 1.

4.4(1).5.5 Table-type MON-NVSI -- Monthly net vertical sediment input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-NVSI
  <-range><---------------mon-nvsi------------------------------------>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-NVSI

*******
Example
*******

  MON-NSVI
    <PLS >  Monthly net vertical sediment input***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7 -.01 -.02 -.03 -.04 -.05 -.03 -.02 -.01  0.0  .01  .03  .01
  END MON-NVSI

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-nvsi>	NVSIM(12)	12F5.0	0.0	none	none	lb / ac.day	Engl
			0.0	none	none	kg / ha.day	Metric

Explanation

This table is only required if VSIVFG in Table-type SED-PARM1 is 1.

4.4(1).5.6 Table-type SED-STOR -- Detached sediment storage

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SED-STOR
  <-range><--------------sed-stor-------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END SED-STOR

*******
Example
*******

  SED-STOR
    <PLS >  Detached sediment storage (tons/acre)    ***
    # -  #    Block1    Block2    Block3    Block4   ***
    1    7       0.2       1.5       4.0       9.0
  END SED-STOR

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sed-stor>	DETSB(5)	5F10.0	0.0	0.0	none	tons/ac	Engl
			0.0	0.0	none	tonnes /ha	Metric

Explanation

DETSB is the initial storage of detached sediment. The system expects a value for each block of the PLS (i.e., NBLKS values), starting with Block no. 1.

4.4(1).6 PERLND BLOCK -- Section PSTEMP input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type PSTEMP-PARM1]
  Table-type PSTEMP-PARM2                Tables in brackets [] are
 [Table-type MON-ASLT]                   not always requried
 [Table-type MON-BSLT]
 [Table-type MON-ULTP1]
 [Table-type MON-ULTP2]
 [Table-type MON-LGTP1]
 [Table-type MON-LGTP2]
 [Table-type PSTEMP-TEMPS]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

4.4(1).6.1 Table-type PSTEMP-PARM1 -- Flags for section PSTEMP

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PSTEMP-PARM1
  <-range><---pstemp-parm1--->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END PSTEMP-PARM1

*******
Example
*******

  PSTEMP-PARM1
    <PLS >  Flags for section PSTEMP***
    # -  # SLTV ULTV LGTV TSOP***
    1    7    0    0    0    1
  END PSTEMP-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<pstemp-parm1>	SLTVFG	4I5	0	0	1
	ULTVFG		0	0	1
	LGTVFG		0	0	1
	TSOPFG		0	0	1

Explanation

If SLTVFG is 1, parameters for estimating surface layer temperature can vary throughout the year. Thus, Table-types MON-ASLT and MON-BSLT will be expected.

ULTVFG serves the same purpose for upper layer temperature calculations. Table-types MON-ULTP1 and MON-ULTP2 will be expected of ULTVFG is 1. LGTVFG serves the same purpose for the lower layer and active groundwater layer temperature calculations. Table-types MON-LGTP1 and MON-LGTP2 will be expected if LGTVFG is 1.

TSOPFG governs the methods used to estimate subsurface soil temperatures. If it is 0, they are computed using a mean departure from air temperature, together with smoothing factors. If TSOPFG is 1, upper layer soil temperature is estimated by regression on air temperature (like surface temperature). The lower layer/groundwater layer temperature is supplied directly by the user (a different value may be specified for each month).

4.4(1).6.2 Table-type PSTEMP-PARM2 -- Second group of PSTEMP parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PSTEMP-PARM2
  <-range><--------------pstemp-parm2------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PSTEMP-PARM2

*******
Example
*******

  PSTEMP-PARM2
    <PLS >***
    # -  #      ASLT      BSLT     ULTP1     ULTP2     LGTP1     LGTP2***
    1    7       24.        .5       24.        .5       40.       0.0
  END PSTEMP-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pstemp-parm2>	ASLT	6F10.0	32.	0.0	100.	deg F	Engl
			0.	-18.	38.	deg C	Metric
	BSLT		1.0	0.001	2.0	deg F/F	Engl
			1.0	0.001	2.0	deg C/C	Metric

Definition of remaining quantities depends on soil temp option flag (TSOPFG in Table-type PSTEMP-PARM1)

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
TSOPFG=0:							
	ULTP1		none	none	none	none	Both
	ULTP2		none	none	none	F deg	Engl
			none	none	none	C deg	Metric
	LGTP1		none	none	none	none	Both
	LGTP2		none	none	none	F deg	Engl
			none	none	none	C deg	Metric
TSOPFG=1:							
	ULTP1		none	none	none	Deg F	Engl
			none	none	none	Deg C	Metric
	ULTP2		none	none	none	Deg F/F	Engl
			none	none	none	Deg C/C	Metric
	LGTP1		none	none	none	Deg F	Engl
			none	none	none	Deg C	Metric
	LGTP2		not used				

Explanation

ASLT is the surface layer temperature, when the air temperature is 32 degrees F (0 degrees C). It is the intercept of the surface layer temperature regression equation. BSLT is the slope of the surface layer temperature regression equation.

If TSOPFG = 0 then:
ULTP1 is the smoothing factor in upper layer temperature calculation.
ULTP2 is the mean difference between upper layer soil temperature and air temperature.
LGTP1 and LGTP2 are the smoothing factor and mean departure from air temperature, for calculating lower layer/groundwater soil temperature.

If TSOPFG = 1 then:
ULTP1 and ULTP2 are the intercept and slope in the upper layer soil temperature regression equation (like ASLT and BSLT for the surface layer). LGTP1 is the lower layer/groundwater layer soil temperature. LGTP2 is not used.

If monthly values are being supplied for any of these quantities (in Table-type MON-XXX), the value appearing in this table is not relevant.

4.4(1).6.3 Table-type MON-ASLT -- Monthly values for ASLT

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-ASLT
  <-range><-----------------mon-aslt---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-ASLT

*******
Example
*******

  MON-ASLT
    <PLS >  Value of ASLT at start of each month (deg F)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  37.  38.  39.  40.  41.  42.  43.  44.  45.  44.  41.  40.
  END MON-ASLT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-aslt>	ASLTM(12)	12F5.0	32. 0.	0. -18.	100. 38.	deg F deg C	Engl Metric

Explanation

This table is only required if SLTVFG in Table-type PSTEMP-PARM1 is 1.

4.4(1).6.4 Table-type MON-BSLT -- Monthly values for BSLT

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-BSLT
  <-range><----------------mon-bslt-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-BSLT

*******
Example
*******

  MON-BSLT
    <PLS >  Value of BSLT at start of each month (deg F/F)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7   .3   .3   .3   .4   .4   .5   .5   .5   .4   .4   .4   .3
  END MON-BSLT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-bslt>	BSLTM(12)	12F5.0	1.0	0.001	2.0	deg F/F	Engl
			1.0	0.001	2.0	deg C/C	Metric

Explanation

This table is only required if SLTVFG in Table-type PSTEMP-PARM1 is 1.

4.4(1).6.5 Table-type MON-ULTP1 Monthly values for ULTP1

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-ULTP1
  <-range><---------------mon-ultp1--------------------------------->
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  (repeats until all operations of this type are covered)
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  END MON-ULTP1

*******
Example
*******


  MON-ULTP1
    <PLS >  Value of ULTP1 at start of each month (TSOPFG=1)          ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  37.  38.  39.  40.  42.  44.  47.  44.  42.  39.  39.  39.
  END MON-ULTP1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<mon-ultp1>	ULTP1M(12)	12F5.0	see notes for Table-type PSTEMP-PARM2		

Explanation

This table is only required if ULTVFG in Table-type PSTEMP-PARM1 is 1.

4.4(1).6.6 Table-type MON-ULTP2 -- Monthly values for ULTP2

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-ULTP2
  <-range><---------------mon-ultp2----------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-ULTP2

*******
Example
*******

  MON-ULTP2
    <PLS >  Value of ULTP2 at start of each month (TSOPFG=1)          ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7   .3   .3   .4   .5   .5   .5   .6   .6   .5   .4   .4   .3
  END MON-ULTP2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<mon-ultp2>	ULTP2M(12)	12F5.0	see notes for Table-type PSTEMP-PARM2		

Explanation

This table is only required if ULTVFG in Table-type PSTEMP-PARM1 is 1.

4.4(1).6.7 Table-type MON-LGTP1 -- Monthly values for LGTP1

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-LGTP1
  <-range><--------------mon-lgtp1-------------------------------->
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  (repeats until all operations of this type are covered)
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  END MON-LGTP1

*******
Example
*******

  MON-LGTP1
    <PLS >  Value of LGTP1 at start of each month (TSOPFG=1)         ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  35.  38.  41.  43.  51.  45.  46.  45.  39.  37.  35.  35.
  END MON-LGTP1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<mon-lgtp1>	LGTP1M(12)	12F5.0	see notes for Table-type PSTEMP-PARM2		

Explanation

This table is only required if LGTVFG in Table-type PSTEMP-PARM1 is 1.

4.4(1).6.8 Table-type MON-LGTP2 -- Monthly values for LGTP2

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-LGTP2
  <-range><--------------mon-lgtp2--------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-LGTP2

*******
Example
*******

  MON-LGTP2
    <PLS >  Value for LGTP2 at start of each month (F deg) (TSOPFG=0) ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  2.0  2.0  2.0  2.0  1.0  1.0  1.0  0.0  0.0  0.0  1.0  2.0
  END MON-LGTP2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-lgtp2>	LGTP2M(12)	12F5.0	none	none	none	F deg	Engl
			none	none	none	C deg	Metric

Explanation

This table is only required if LGTVFG in Table-type PSTEMP-PARM1 is 1 and TSOPFG is 0.

4.4(1).6.9 Table-type PSTEMP-TEMPS -- Initial temperatures

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PSTEMP-TEMPS
  <-range><------------pstemp-temps-------------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END PSTEMP-TEMPS

*******
Example
*******

  PSTEMP-TEMPS
    <PLS >  Initial temperatures***
    # -  #     AIRTC     SLTMP     ULTMP     LGTMP***
    1    7       48.       48.       48.       48.
  END PSTEMP-TEMPS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pstemp-temps>	AIRTC	4F10.0	60.	-20.	120.	deg F	Engl
			16.	-29.	49.	deg C	Metric
	SLTMP		60.	-20.	120.	deg F	Engl
			16.	-29.	49.	deg C	Metric
	ULTMP		60.	-20.	120.	deg F	Engl
			16.	-29.	49.	deg C	Metric
	LGTMP		60.	-20.	120.	deg F	Engl
			16.	-29.	49.	deg C	Metric

Explanation

These are the initial temperatures:

AIRTC - air temperature
SLTMP - surface layer soil temperature
ULTMP - upper layer soil temperature
LGTMP - lower layer/groundwater layer soil temperature

4.4(1).7 PERLND BLOCK -- Section PWTGAS input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type PWT-PARM1]
 [Table-type PWT-PARM2]                   Tables in brackets [] are not
 [Table-type MON-IFWDOX]                  always required
 [Table-type MON-IFWCO2]
 [Table-type MON-GRNDDOX]
 [Table-type MON-GRNDCO2]
 [Table-type PWT-TEMPS]
 [Table-type PWT-GASES]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

4.4(1).7.1 Table-type PWT-PARM1 — Flags for section PWTGAS

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWT-PARM1
  <-range><----pwt-parm1----->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END PWT-PARM1

*******
Example
*******

  PWT-PARM1
    <PLS >  Flags for section PWTGAS***
    # -  #  IDV  ICV  GDV  GVC***
    1    7    0    0    1    0
  END PWT-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<pwt-parm1>	IDVFG	4I5	0	0	1
	ICVFG		0	0	1
	GDVFG		0	0	1
	GCVFG		0	0	1

Explanation

These flags each indicate whether or not a parameter is allowed to vary throughout the year and, thus, whether or not the corresponding table of monthly values will be expected:

FLAG	PARAMETER	TABLE-TYPE FOR MONTHLY VALUES
IDVFG	Interflow DO concentration	MON-IFWDOX
ICVFG	Interflow CO2 concentration	MON-IFWCO2
GDVFG	Groundwater DO concentration	MON-GRNDDOX
GCVFG	Groundwater CO2 concentration	MON-GRNDCO2

4.4(1).7.2 Table-type PWT-PARM2 -- Second group of PWTGAS parms

```
****************************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
****************************************************************************************
Layout
******

  PWT-PARM2
  <-range><-----------------pwt-parm2--------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PWT-PARM2

*******
Example
*******

  PWT-PARM2
    <PLS >  Second group of PWTGAS parms***
    # -  #      ELEV     IDOXP     ICO2P     ADOXP     ACO2P***
    1    7     1281.       8.2       0.2       8.2       0.3
  END PWT-PARM2

****************************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwt-parm2>	ELEV	5F10.0	0.0	-1000.	30000.	ft	Engl
			0.0	-300.	9100.	m	Metric
	DOXP		0.0	0.0	20.	mg/l	Both
	ICO2P		0.0	0.0	1.0	mg C/l	Both
	ADOXP		0.0	0.0	20.	mg/l	Both
	ACO2P		0.0	0.0	1.0	mg C/l	Both

Explanation

ELEV is the elevation of the PLS above sea level (used to adjust saturation concentrations of dissolved gasses in surface outflow).

IDOXP is the concentration of dissolved oxygen in interflow outflow.
ICO2P is the concentration of dissolved CO2 in interflow outflow.
ADOXP is the concentration of dissolved oxygen in active groundwater outflow.
ACO2P is the concentration of dissolved CO2 in active groundwater outflow.

4.4(1).7.3 Table-type MON-IFWDOX -- Monthly interflow DO concentration

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-IFWDOX
  <-range><----------------mon-ifwdox-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-IFWDOX

*******
Example
*******

  MON-IFWDOX
    <PLS >  Value at start of each month for interflow DO concentration***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  4.5  4.7  5.7  6.5  7.6  7.6  7.4  6.3  4.3  5.3  4.3  3.5
  END MON-IFWDOX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-ifwdox>	IDOXPM(12)	12F5.0	0.0	0.0	20.0	mg/l	Both

Explanation

This table is only required if IDVFG in Table-type PWT-PARM1 is 1.

4.4(1).7.4 Table-type MON-IFWCO2 — Monthly interflow CO2 concentration

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-IFWCO2
  <-range><--------------mon-ifwco2-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-IFWCO2

*******
Example
*******

  MON-IFWCO2
    <PLS >  Value at start of each month for interflow CO2 concentration***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7 .123 .171 .142 .145 .157 .178 .122 .123 .143 .145 .176 .145
  END MON-IFWCO2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-ifwco2>	ICO2PM(12)	12F5.0	0.0	0.0	1.0	mg C/l	Both

Explanation

This table is only required if ICVFG in Table-type PWT-PARM1 is 1.

4.4(1).7.5 Table-type MON-GRNDDOX -- Monthly groundwater DO concentration

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-GRNDDOX
  <-range><---------------mon-grnddox-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-GRNDDOX

*******
Example
*******

  MON-GRNDDOX
    <PLS >  Value at start of each month for groundwater DO concentration***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  4.5  4.7  4.9  4.9  4.9  4.9  5.0  5.6  5.7  5.8  5.4  5.1
  END MON-GRNDDOX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-grnddox>	ADOXPM(12)	12F5.0	0.0	0.0	20.0	mg/l	Both

Explanation

This table is only required if GDVFG in Table-type PWT-PARM1 is 1.

4.4(1).7.6 Table-type MON-GRNDCO2 — Monthly groundwater CO2 concentration

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-GRNDCO2
  <-range><--------------mon-grndco2-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-GRNDCO2

*******
Example
*******

  MON-GRNDCO2
    <PLS >  Value at start of each month for groundwater CO2 concentration***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .23  .22  .22  .23  .24  .25  .24  .23  .22  .22  .22  .22
  END MON-GRNDCO2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-grndco2>	ACO2PM(12)	12F5.0	0.0	0.0	1.0	mg C/l	Both

Explanation

This table is only required if GCVFG in Table-type PWT-PARM1 is 1.

4.4(1).7.7 Table-type PWT-TEMPS — Initial water temperatures

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWT-TEMPS
  <-range><----------pwt-temps--------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END PWT-TEMPS

*******
Example
*******

  PWT-TEMPS
    <PLS >    Initial water temperatures***
    # -  #     SOTMP     IOTMP     AOTMP***
    1    7       47.       47.       53.
  END PWT-TEMPS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwt-temps>	SOTMP	3F10.0	60.	32.	100.	deg F	Engl
			16.	0.	38.	deg C	Metric
	IOTMP		60.	32.	100.	deg F	Engl
			16.	0.	38.	deg C	Metric
	AOTMP		60.	32.	100.	deg F	Engl
			16.	0.	38.	deg C	Metric

Explanation

These are the initial water temperatures:

SOTMP is surface outflow temperature.
IOTMP is interflow outflow temperature.
AOTMP is active groundwater outflow temperature.

4.4(1).7.8 Table-type PWT-GASES -- Initial DO and CO2 concentrations

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PWT-GASES
  <-range><---------------pwt-gases--------------------------------->
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  (repeats until all operations of this type are covered)
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  END PWT-GASES

*******
Example
*******

  PWT-GASES
    <PLS >          Initial DO and CO2 concentrations***
    # -  #     SODOX     SOCO2     IODOX     IOCO2     AODOX     AOCO2***
    1    7       8.9      .122       7.8      .132       3.5      .132
  END PWT-GASES

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<pwt-gases>	SODOX	6F10.0	0.0	0.0	20.	mg/l	Both
	SOCO2		0.0	0.0	1.0	mg C/l	Both
	IODOX		0.0	0.0	20.	mg/l	Both
	IOCO2		0.0	0.0	1.0	mg C/l	Both
	AODOX		0.0	0.0	20.	mg/l	Both
	AOCO2		0.0	0.0	1.0	mg C/l	Both

Explanation

These are the initial concentrations of dissolved gas:

SODOX is DO concentration in surface outflow.
SOCO2 is CO2 concentration in surface outflow.
IODOX is DO concentration in interflow outflow.
IOCO2 is CO2 concentration in interflow outflow.
AODOX is DO concentration in active groundwater outflow.
AOCO2 is CO2 concentration in active groundwater outflow.

4.4(1).8 PERLND BLOCK — Section PQUAL input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type NQUALS]
                                --
  Table-type QUAL-PROPS           |
 [Table-type QUAL-INPUT]          |
 [Table-type MON-POTFW]           |
 [Table-type MON-POTFS]           | repeat for each
 [Table-type MON-ACCUM]           | quality constituent
 [Table-type MON-SQOLIM]          |
 [Table-type MON-IFLW-CONC]       |
 [Table-type MON-GRND-CONC]       |
                                --
********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] are not always required;for example, because all the values can be defaulted.

4.4(1).8.1 Table-type NQUALS -- Total number of quality constituents simulated

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NQUALS
  <-range><nql>
  . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . .
  END NQUALS

*******
Example
*******

  NQUALS
    <PLS >     ***
    # -  #NQUAL***
    1    7    8
  END NQUALS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<nql>	NQUAL	I5	1	1	10

Explanation

The total number of quality constitutents simulated in Section PQUAL is indicated in this table. The set of tables below is repeated for each quality constitutent (but any tables not applicable to a given constituent may be omitted).

4.4(1).8.2 Table-type QUAL-PROPS -- Identifiers and Flags for a quality constituent

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  QUAL-PROPS
  <-range><-qualid->      <qt><------------------flags--------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END QUAL-PROPS

*******
Example
*******

  QUAL-PROPS
    <PLS >  Identifiers and Flags***
    # -  #*** qualid      QTID  QSD VPFW VPFS  QSO  VQO QIFW VIQC QAGW VAQC
    1    7       BOD       kg    0    0    0    1    1    1    0    1    1
  END QUAL-PROPS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<qualid>	QUALID	5A2	none	none	none
<qt>	QTYID	2A2	none	none	none
<flags>	QSDFG	9I5	0	0	1
	VPFWFG		0	0	1
	VPFSFG		0	0	1
	QSOFG		0	0	1
	VQOFG		0	0	1
	QIFWFG		0	0	1
	VIQCFG		0	0	1
	QAGWFG		0	0	1
	VAQCFG		0	0	1

Explanation

QUALID is a string of up to 10 characters which identifies the quality constituent. QTYID is a string of up to 4 characters which identifies the units associated with this constituent (e.g., kg, # (for coliforms)). These are the units refered to as "qty" in susequent tables (eg. Table-type QUAL-INPUT).

If QSDFG is 1 then:
1. This constituent is a QUALSD (sediment associated).
2. If VPFWFG is 1, the washoff potency factor may vary throughout the year. Table-type MON-POTFW is expected.
3. If VPFSFG is 1, the scour potency factor may vary throughout the year. Table-type MON-POTFS is expected.

If QSOFG is 1 then:
1. This constituent is a QUALOF (directly associated with overland flow).
2. If VQOFG is 1 then rate of accumulation and the limiting storage of QUALOF may vary throughout the year. Table-types MON-ACCUM and MON-SQOLIM are expected.

If QIFWFG is 1 then:
1. This constituent is a QUALIF (interflow associated).
2. If VIQCFG is 1 then concentration of this constituent in interflow outflow may vary throughout the year. Table-type MON-IFLW-CONC is expected.

If QAGWFG is 1 then:
1. This constituent is a QUALGW (groundwater associated).
2. If VAQCFG is 1 the concentration of this constituent in groundwater outflow may vary throughout the year. Table-type MON-GRND-CONC is expected.

4.4(1).8.3 Table-type QUAL-INPUT -- Storage on surface and nonseasonal parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  QUAL-INPUT
  <-range><---------------qual-input------------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END QUAL-INPUT

*******
Example
*******

  QUAL-INPUT
    <PLS >  Storage on surface and nonseasonal parameters***
    # -  #     SQO   POTFW   POTFS   ACQOP  SQOLIM   WSQOP    IOQC    AOQC***
    1    7    1.21    17.2     1.1    0.02     2.0    1.70    15.2    17.1
  END QUAL-INPUT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<qual-input>	SQO	8F8.0	0.0	0.0	none	qty/ac	Engl
			0.0	0.0	none	qty/ha	Metric
	POTFW		0.0	0.0	none	qty/ton	Engl
			0.0	0.0	none	qty /tonne	Metric
	POTFS		0.0	0.0	none	qty/ton	Engl
			0.0	0.0	none	qty /tonne	Metric
	ACQOP		0.0	0.0	none	qty /ac.day	Engl
			0.0	0.0	none	qty /ha.day	Metric
	SQOLIM		0.01	0.01	none	qty/ac	Engl
			0.02	0.02	none	qty/ha	Metric
	WSQOP		1.64	0.01	none	in/hr	Engl
			41.7	0.25	none	mm/hr	Metric
	IOQC		0.0	0.0	none	qty/ft3	Engl
			0.0	0.0	none	qty/l	Metric
	AOQC		0.0	0.0	none	qty/ft3	Engl
			0.0	0.0	none	qty/l	Metric

Explanation

The following variables are applicable only if the constituent is a QUALSD:
1. POTFW, the washoff potency factor.
2. POTFS, the scour potency factor.

A potency factor is the ratio of constituent yeild to sediment (washoff or scour) outflow.

The following variables are applicable only if the constituent is a QUALOF:
1. SQO, the initial storage of QUALOF on the surface of the PLS.
2. ACQOP, the rate of accumulation of QUALOF.
3. SQOLIM, the maximum storage of QUALOF.
4. WSQOP, the rate of surface runoff which will remove 90 percent of stored QUALOF per hour.

IOQC is the concentration of the constituent in interflow outflow (meaningful only if this is a QUALIF). AOQC is the concentration of the constituent in active groundwater outflow (meaningful only if this is a QUALGW).

If monthly values are being supplied for any of these quantities, the value in this table is not relevant; instead,the system expects and uses values supplied in Table-type MON-XXX.

4.4(1).8.4 Table-type MON-POTFW -- Monthly washoff potency factor

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-POTFW
  <-range><--------------mon-potfw------------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-POTFW

*******
Example
*******

  MON-POTFW
    <PLS >  Value at start of each month for washoff potency factor (lb/ton)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  1.2  2.4  3.6  5.8 10.2 20.2 25.2 30.8 40.2 10.1  2.5  1.7
  END MON-POTFW

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-potfw>	POTFWM(12)	12F5.0	0.0	0.0	none	qty/ton	Engl
			0.0	0.0	none	qty /tonne	Metric

Explanation

This table is only required if VPFWFG in Table-type QUAL-PROPS is 1.

4.4(1).8.5 Table-type MON-POTFS -- Monthly scour potency factor

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-POTFS
  <-range><--------------mon-potfs---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-POTFS

*******
Example
*******

  MON-POTFS
    <PLS >  Value at start of each month for scour potency factor (lb/ton)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  0.9  0.9  0.9  0.8  0.8  1.1  1.1  1.3  1.3  1.0  0.9  0.9
  END MON-POTFS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-potfs>	POTFSM(12)	12F5.0	0.0	0.0	none	qty/ton	Engl
			0.0	0.0	none	qty /tonne	Metric

Explanation

This table is only required if VPFSFG in Table-type QUAL-PROPS is 1.

4.4(1).8.6 Table-type MON-ACCUM -- Monthly accumulation rates of QUALOF

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-ACCUM
  <-range><--------------mon-accum-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-ACCUM

*******
Example
*******

  MON-ACCUM
    <PLS >  Value at start of month for accum rate of QUALOF (lb/ac.day)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  0.0  0.0 0.01 0.02 0.02 0.04 0.05 0.04 0.02 0.01  0.0  0.0
  END MON-ACCUM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-accum>	ACQOPM(12)	12F5.0	0.0	0.0	none	qty /ac.day	Engl
			0.0	0.0	none	qty /ha.day	Metric

Explanation

This table is only required if VQOFG in Table-type QUAL-PROPS is 1.

4.4(1).8.7 Table-type MON-SQOLIM -- Monthly limiting storage of QUALOF

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-SQOLIM
  <-range><-----------------mon-sqolim------------------------------>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-SQOLIM

*******
Example
*******

  MON-SQOLIM
    <PLS >  Value at start of month for limiting storage of QUALOF (lb/acre)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7   10   12   14   18   20   25   30   26   20   13   10    7
  END MON-SQOLIM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-sqolim>	SQOLIM(12)	12F5.0	none	0.01	none	qty/ac	Engl
			none	0.02	none	qty/ha	Metric

Explanation

This table is only required if VQOFG in Table-type QUAL-PROPS is 1.

4.4(1).8.8 Table-type MON-IFLW-CONC -- Monthly conc of QUAL in interflow

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-IFLW-CONC
  <-range><----------------mon-iflw-conc---------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-IFLW-CONC

*******
Example
*******

  MON-IFLW-CONC
    <PLS >  Conc of QUAL in interflow outflow for each month (lb/ft3)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1     7.0012.0010.0005  0.0  0.0.0002 .005 .002 .001.0016.0014.0012
  END MON-IFLW-CONC

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-iflw-conc>	IOQCM(12)	12F5.0	0.0	0.0	none	qty/ft3	Engl
			0.0	0.0	none	qty/l	Metric

Explanation

This table is only required if VIQCFG in Table-type QUAL-PROPS is 1.

4.4(1).8.9 Table-type MON-GRND-CONC -- Monthly conc of QUAL in groundwater

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-GRND-CONC
  <-range><---------------mon-grnd-conc------------------------------>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-GRND-CONC

*******
Example
*******

  MON-GRND-CONC
    <PLS >  Value at start of month for conc of QUAL in groundwater (lb/ft3)**
    # - #   JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7.0013.0014.0012.0012.0012.001 .001 .001 .0011.0012.0012.0013
  END MON-GRND-CONC

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-grnd-conc>	AOQCM(12)	12F5.0	0.0	0.0	none	qty/ft3	Engl
			0.0	0.0	none	qty/l	Metric

Explanation

This table is only required if VAQCFG in Table-type QUAL-PROPS is 1.

4.4(1).9 PERLND BLOCK — Section MSTLAY input

Layout:

```
                                        --
  Table-type VUZFG                        | if Section
  Table-type UZSN-LZSN                    | PWATER is
  Table-type MON-UZSN if VUZFG= 1         | inactive
                                        --
  Table-type MST-PARM
                                        --
  Table-type MST-TOPSTOR                  | repeat for
  Table-type MST-TOPFLX                   | each block
                                        --
  Table-type MST-SUBSTOR
  Table-type MST-SUBFLX
```

Explanation:

The exact format of each of the tables mentioned above, except MON-UZSN, is detailed in the documentation which follows. MON-UZSN is documented under the input for Section PWATER (4.4(1).4).

The comments given alongside the table names above indicate:

1. Under which circumstances a table is expected
2. Sequencing information. Note that "repeat for each block" means that the bracketed set of tables is repeated for each areal source block in the pervious land-segment (PLS). The first set is for block 1, second for block 2, etc. The number of blocks (NBLKS) was specified in Table-type GEN-INFO (Sect. 4.4(1).1.3).

Note that if all the fields in a table have default values, the table can be omitted from the User's Control Input. Then, the defaults will be adopted.

Table-types MST-TOPSTOR through MST-SUBFLX should usually not be supplied. See the documentation of those tables for further details.

4.4(1).9.1 Table-type VUZFG — Variable upper zone flag

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  VUZFG
  <-range><vuz>
  . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . .
  END VUZFG

*******
Example
*******

  VUZFG
    <PLS >VUZFG***
    # -  #     ***
    1    7    1
  END VUZFG

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<vuz>	VUZFG	I5	0	0	1

Explanation

VUZFG is a flag which indicates whether or not the upper zone nominal storage varies throughout the year or not. A value of zero means it does not vary, value 1 means it does. If it does vary, the system will expect a table of type MON-UZSN in the User's Control Input.

Note that Table VUZFG is only required if Section PWATER is inactive. If that section is active VUZFG would have already been provided in the input for PWATER (Table-type PWAT-PARM1).

4.4(1).9.2 Table-type UZSN-LZSN — Values of UZSN, LZSN and initial surface storages

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  UZSN-LZSN
  <-range><-uzsn-><-lzsn-><-surs-><--------------sursb------------------>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END UZSN-LZSN

*******
Example
*******

  UZSN-LZSN
    <PLS >     UZSN     LZSN     SURS       SURSB       ***
    # -  #       in       in       in  Block1  Block2***
    1    7      1.0      6.0      .02     .01     .03
  END UZSN-LZSN

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<uzsn>	UZSN	F8.0	none	0.01	10.0	in	Engl
			none	0.25	250.	mm	Metric
<lzsn>	LZSN	F8.0	none	0.01	100.	in	Engl
			none	0.25	2500.	mm	Metric
<surs>	SURS	F8.0	.001	.001	100.	in	Engl
			.025	.025	2500.	mm	Metric
<sursb>	SURSB(*)	F8.0	.001	.001	100.	in	Engl
			.025	.025	2500.	mm	Metric

Explanation

This table is only required if Section PWATER is inactive, else the data would have already been supplied in the input for Section PWATER.

UZSN is the nominal upper zone storage. The value supplied here is irrelevant if VUZFG has been set to 1; in that case monthly values for UZSN are supplied in Table-type MON-UZSN.

LZSN is the nominal lower zone storage.

SURS is the initial surface detention storage. If the PLS consists of more than one areal source "block", this value should be the mean of the values given for each block.

SURSB(*) are the initial surface detention storages on each block of the PLS. A value should be supplied for each block, except if there is only one block, in which case no value need be supplied.

4.4(1).9.3 Table-type MST-PARM -- Factors used to adjust solute leaching rates

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MST-PARM
  <-range><--------leach-parms--------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END MST-PARM

*******
Example
*******

  MST-PARM
    <PLS >     SLMPF      ULPF      LLPF***
    # -  #                              ***
    1    7       0.5       2.0       2.0
  END MST-PARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<leach-parms>	SLMPF	3F10.0	1.0	.001	1.0	none	Both
	ULPF		1.0	1.0	5.0	none	Both
	LLPF		1.0	1.0	5.0	none	Both

Explanation

These are the factors used to adjust solute percolation rates. SLMPF affects percolation from the surface layer storage to the upper layer principal storage. ULPF affects percolation from the upper layer principal storage to the lower layer storage. LLPF affects percolation from the lower layer storage to the active and inactive groundwater.

4.4(1).9.4 Table-type MST-TOPSTOR -- Initial moisture storage in each topsoil layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MST-TOPSTOR
  <-range><---------topstor------------>
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END MST-TOPSTOR

*******
Example
*******

  MST-TOPSTOR
    <PLS >      Topsoil storages (lb/ac)***
    # -  #     SMSTM     UMSTM     IMSTM***
    1    7    100000    400000    300000
  END MST-TOPSTOR

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<topstor>	SMSTM	3F10.0	0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric
	UMSTM		0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric
	IMSTM		0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric

Explanation

This table is used to specify the initial moisture content in the surface, upper principal and upper transitory (interflow) storages respectively. The entire table should be repeated for each block of the PLS; block1 first, etc.

Note that the values given in this table only affect the water storages for the start of the first interval in the run; there is no carry-over of the values beyond the starting instant. Therefore, in most runs, this table need not be supplied; the default zero values will not cause any problems.

4.4(1).9.5 Table-type MST-TOPFLX -- Initial fractional fluxes in topsoil layers

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MST-TOPFLX
  <-range><------------------top-flux------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MST-TOPFLX

*******
Example
*******

  MST-TOPFLX
    <PLS >   Fractional fluxes in topsoil layers (/ivl)     ***
    # -  #        FSO       FSP       FII       FUP       FIO***
    1    7        .07       .03
  END MST-TOPFLX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<top-flux>	FSO,FSP,FII, FUP,FIO	5F10.0	0.0	0.0	1.0	/ivl	Both

Explanation

These are the initial values of the fractional fluxes of soluble chemicals through the topsoil layers of a PLS. If the PLS is subdivided into more than one block, a separate table should be supplied for each block; block1 first, etc

Note that the values supplied in this table apply at the instant that the run starts. The program computes new values each time step and there is no carry-over of values from one time step to the next. Therefore, in most runs, you can omit this table; the default zero values will not cause any problems.

4.4(1).9.6 Table-type MST-SUBSTOR -- Initial moisture storage in subsurface layers

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MST-SUBSTOR
  <-range><-----substor------>
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END MST-SUBSTOR

*******
Example
*******

  MST-SUBSTOR
     <PLS >Subsoil moisture (kg/ha)***
    # -  #     LMSTM     AMSTM    ***
    1    7    800000   1000000
  END MST-SUBSTOR

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<substor>	LMSTM,AMSTM	2F10.0	0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric

Explanation

These are the initial moisture storages in the lower layer and active groundwater layers respectively.

Usually, this table should be omitted and the default values taken. The comments made on this subject in the explanation for Table-type MST-TOPSTOR are also applicable here.

4.4(1).9.7 Table-type MST-SUBFLX -- Initial fractional fluxes in subsurface layers

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MST-SUBFLX
  <-range><----------subflux----------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END MST-SUBFLX

*******
Example
*******

  MST-SUBFLX
    <PLS >Subsurface fractional fluxes (/ivl) ***
    # -  #       FLP      FLDP       FAO      ***
    1    7       0.1      0.05
  END MST-SUBFLX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<subflux>	FLP,FLDP,FAO	3F10.0	0.0	0.0	1.0	/ivl	Both

Explanation

These are the initial fractional fluxes of soluble chemicals through the subsoil layers.

Usually, this table should be omitted and the default values taken. The comments on this subject in the explanation for Table-type MST-TOPFLX are applicable here.

4.4(1).10 PERLND BLOCK — Section PEST input

Layout:

```
  Table-type PEST-FLAGS
  Table-type SOIL-DATA
                                                 ------------------
  Table-type PEST-ID                                               ¦
                                                 --                ¦
  Table-type PEST-THETA                            ¦               ¦
  Table-type PEST-FIRSTPM   for surface layer      ¦ if            ¦
  Table-type PEST-FIRSTPM   for upper layer        ¦ ADOPFG        ¦
  Table-type PEST-FIRSTPM   for lower layer        ¦ =1            ¦
  Table-type PEST-FIRSTPM   for groundwater layer  ¦               ¦
                                                 --                ¦
                                                 --                ¦
  Table-type PEST-CMAX                             ¦               ¦
  Table-type PEST-SVALPM    for surface layer      ¦ if            ¦
  Table-type PEST-SVALPM    for upper layer        ¦ ADOPFG        ¦
  Table-type PEST-SVALPM    for lower layer        ¦ =2            ¦
  Table-type PEST-SVALPM    for groundwater layer  ¦               ¦
                                                 --                ¦repeat for
                                                 --                ¦each
  Table-type PEST-CMAX                             ¦               ¦pesticide
  Table-type PEST-NONSVPM   for surface layer      ¦ if            ¦
  Table-type PEST-NONSVPM   for upper layer        ¦ ADOPFG        ¦
  Table-type PEST-NONSVPM   for lower layer        ¦ =3            ¦
  Table-type PEST NONSVPM   for groundwater layer  ¦               ¦
                                                 --                ¦
  Table-type PEST-DEGRAD                                           ¦
                                                      --           ¦
  Table-type PEST-STOR1 for surface layer storage       ¦repeat for¦
  Table-type PEST-STOR1 for upper layer princ. storage  ¦each block¦
  Table-type PEST-STOR2 for upper layer trans. storage  ¦          ¦
                                                      --           ¦
  Table-type PEST-STOR1 for lower layer storage                    ¦
  Table-type PEST-STOR1 for groundwater layer storage              ¦
                                                 ------------------
```

Explanation:

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

The comments given alongside the table names above indicate:

1. Under which circumstances a table is expected
2. Sequencing information. Note that "repeat for each block" means that the bracketed set of tables is repeated for each areal source block in the pervious land-segment (PLS). The first set is for block 1, second for block 2, etc. The number of blocks (NBLKS) was specified in Table-type GEN-INFO (Sect. 4.4(1).1.3).

Note that if all the fields in a table have default values, the table can be omitted from the User's Control Input. Then, the defaults will be adopted.

ADOPFG is the adsorption/desorption option flag. It is described in the documentation for Table-type PEST-FLAGS (Sect. 4.4(1).10.1) below.

4.4(1).10.1 Table-type PEST-FLAGS -- Flags for pesticide simulation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-FLAGS
  <-range><nps><----itmax----><----adopt---->
  . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . .
  END PEST-FLAGS

*******
Example
*******

  PEST-FLAGS
    <PLS > NPST|Max iterations|Adsorp option ***
    # -  #     |Pst1 Pst2 Pst3|Pst1 Pst2 Pst3***
    1    7    2   20   20         1    3
  END PEST-FLAGS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<nps>	NPST	I5	1	1	3
<itmax>	ITMXPS(*)	3I5	30	1	100
<adopt>	ADOPFG(*)	3I5	2	1	3

Explanation

NPST is the number of pesticides being simulated in the operation.

ITMXPS is the maximum number of iterations that will be made in trying to solve for adsorbed and dissolved equilibrium using the Freundlich isotherm. A separate value may be supplied for each pesticide being handled (up to 3). If the Freundlich method is not being used, these values have no effect.

ADOPFG(*) are flags which indicate which method will be used to simulate adsorption/desorption, for each pesticide (maximum of 3):

1 means use first-order kinetics
2 means use single-value Freundlich method
3 means use non-single value Freundlich method

4.4(1).10.2 Table-type SOIL-DATA -- Soil layer depths and bulk densities

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SOIL-DATA
  <-range><------------depths------------><-----------bulkdens----------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END SOIL-DATA

*******
Example
*******

  SOIL-DATA
    <PLS >|           Depths (ins)         |    Bulk density (lb/ft3)      |***
    # -  #|Surface   Upper   Lower Groundw|Surface   Upper   Lower Groundw|***
    1    7     .12     6.0    40.0     80.    80.                     120.
  END SOIL-DATA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<depths>	none	4F8.0	none	.001	1000	in	Engl
			none	.0025	2500	cm	Metric
<bulkdens>	none	4F8.0	103	50	150	lb/ft3	Engl
			1.65	0.80	2.40	gm/cc	Metric

Explanation

The first four values are the depths (thicknesses) of the surface, upper, lower and groundwater layers respectively; the second group of four values are the corresponding bulk densities of the soil in those layers.

The depth and bulk density are mutiplied together by the program to obtain the mass of soil in each layer. This is used to compute the concentrations of adsorbed chemicals.

4.4(1).10.3 Table-type PEST-ID — Name of pesticide

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-ID
  <-range><------pestid------>
  . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . .
  END PEST-ID

*******
Example
*******

  PEST-ID
    <PLS >               Pesticide***
    # -  #                       ***
    1    7                Atrazine
  END PEST-ID

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<pestid>	PESTID(*)	5A4	none	none	none

Explanation

This table specifies the name of the pesticide to which the data in the following tables apply.

4.4(1).10.4 Table-type PEST-THETA -- Pesticide first-order reaction temperature correction parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-THETA
  <-range><------theta------->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END PEST-THETA

*******
Example
*******

  PEST-THETA
    <PLS >   Temperature parms***
    # -  #    THDSPS    THADPS***
    1    7                1.07
  END PEST-THETA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<theta>	THDSPS,THADPS	2F10.0	1.05	1.00	2.00	none	Both

Explanation

These parameters are used to adjust the desorption and adsorption rate parameters (respectively), using a modified Arrhenius equation:

`Rate at T= (Rate at 35degC)* (theta)**(T-35)`

This table is only required if first order kinetics are used to simulate adsorption/desorption (ADOPFG=1 in Table-type PEST-FLAGS)

4.4(1).10.5 Table-type PEST-FIRSTPM -- Pesticide first-order parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-FIRSTPM
  <-range><----firstparm----->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END PEST-FIRSTPM

*******
Example
*******

  PEST-FIRSTPM
    <PLS >First-order parms (/day)***
    # -  #     KDSPS     KADPS    ***
    1    7       .07       .04
  END PEST-FIRSTPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<firstparm>	KDSPS,KADPS	2F10.0	0.0	0.0	none	/day	Both

Explanation

KDSPS and KADPS are the desorption and adsorption rates at 35degC.

This table is only required if ADOPFG=1 (first-order kinetics) for this pesticide.

4.4(1).10.6 Table-type PEST-CMAX — Maximum solubility of pesticide

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-CMAX
  <-range><--cmax-->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END PEST-CMAX

*******
Example
*******

  PEST-CMAX
    <PLS >      CMAX***
    # -  #     (ppm)***
    1    7      25.0
  END PEST-CMAX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<cmax>	CMAX	F10.0	0.0	0.0	none	ppm	Both

Explanation

CMAX is the maximum solubility of the pesticide in water.

This table is only required if ADOPFG= 2 or 3 for this pesticide (Freundlich method of simulating adsorption/desorption).

4.4(1).10.7 Table-type PEST-SVALPM -- Pesticide parameters for single value Freundlich method

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-SVALPM
  <-range><----------svalpm----------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END PEST-SVALPM

*******
Example
*******

  PEST-SVALPM
    <PLS >      XFIX        K1         N1***
    # -  #     (ppm)                     ***
    1    7       20.       4.0        1.5
  END PEST-SVALPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<svalpm>	XFIX	3F10.0	0.0	0.0	none	ppm	Both
	K1		0.0	0.0	none		Both
	N1		none	1.0	none		Both

Explanation

XFIX is the maximum concentration (on the soil) of pesticide which is permanently fixed to the soil. K1 and N1 arc the coeff. and exponent parameters for the Freundlich adsorption/desorption equation:

```
X= K1*C**(1/N1) + XFIX
```

This table is only used if ADOPFG= 2 for this pesticide (single value Freundlich method). Then, the system expects it to appear four times for this pesticide; first, for the surface layer, second for the upper layer, etc.

4.4(1).10.8 Table-type PEST-NONSVPM -- Pesticide parameters for Non-single Value Freundlich method

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-NONSVPM
  <-range><---------------nonsvpm-------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END PEST-NONSVPM

*******
Example
*******

  PEST-NONSVPM
    <PLS >      XFIX        K1        N1        N2***
    # -  #     (ppm)                              ***
    1    7       15.       5.0       1.5       1.7
  END PEST-NONSVPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<nonsvpm>	XFIX	4F10.0	0.0	0.0	none	ppm	Both
	K1		0.0	0.0	none		Both
	N1		none	1.0	none		Both
	N2		none	1.0	none		Both

Explanation

XFIX is the maximum concentration (on the soil) of pesticide which is permanently fixed in the soil. K1 and N1 are the coefficient and exponent parameters for the Freundlich curve used for adsorption. N2 is the exponent parameter for the auxiliary ("desorption") curve.

This table is only used if ADOPFG= 3 for this pesticide (Non-single Value Freundlich Method).

4.4(1).10.9 Table-type PEST-DEGRAD -- Pesticide degradation rates

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-DEGRAD
  <-range><----------------degrad---------------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END PEST-DEGRAD

*******
Example
*******

  PEST-DEGRAD
    <PLS >    Pesticide degradation rates (/day)  ***
    # -  #   Surface     Upper     Lower    Groundw***
    1    7       .05       .02       .01
  END PEST-DEGRAD

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<degrad>	SDGCON,UDGCON, LDGCON,ADGCON	4F10.0	0.0	0.0	1.0	/day	Both

Explanation

These are the degradation rates of the pesticide in the surface, upper, lower and groundwater layers respectively. These rates are not adjusted for temperature.

4.4(1).10.10 Table-type PEST-STOR1 -- Initial pesticide storage in surface, upper, lower or groundwater layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-STOR1
  <-range><-cryst--><---ads--><--soln-->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END PEST-STOR1

*******
Example
*******

  PEST-STOR1
    <PLS >Initial pesticide in surface layer (lb/ac)***
    # -  #     Cryst       Ads       Soln            ***
    1    7      10.0      25.0       50.0
  END PEST-STOR1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<cryst>,<ads>, <soln>	PSCY,PSAD, PSSU	3F10.0	0.0 0.0	0.0 0.0	none none	lb/ac kg/ha	Engl Metric

Explanation

<cryst> is the pesticide in crystalline form, <ads> is the pesticide in adsorbed form and <soln> is the pesticide in solution.

The values given in this table apply to one of the following four soil storages: surface, upper principal, lower or groundwater. In the case of the surface and upper principal storages these values may apply to the entire layer (if NBLKS= 1) or to a single block in the layer (if NBLKS> 1).

4.4(1).10.11 Table-type PEST-STOR2 -- Initial pesticide stored in upper layer transitory (interflow) storage

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PEST-STOR2
  <-range><--ips--->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END PEST-STOR2

*******
Example
*******

  PEST-STOR2
    <PLS > Interflow       ***
    # -  #   storage(kg/ha)***
    1    7      20.0
  END PEST-STOR2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ips>	IPS	F10.0	0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric

Explanation

IPS is the initial storage of pesticide in the upper layer transitory (interflow) storage. Since only dissolved pesticide is modeled in that storage, only one value is needed (no crystalline or adsorbed material).

This value may apply to the entire layer (if NBLKS= 1) or to a single block in the layer (if NBLKS> 1).

4.4(1).11 PERLND BLOCK -- Section NITR input

Layout:

```
  Table-type SOIL-DATA  if section PEST is inactive
  Table-type NIT-FLAGS
  Table-type NIT-UPTAKE ----------------------- if VNUTFG= 0
                                              --
  Table-type MON-NITUPT for surface layer     | if VNUTFG= 1
  Table-type MON-NITUPT for upper layer       |
  Table-type MON-NITUPT for lower layer       |
  Table-type MON-NITUPT for groundwater layer |
                                              --
  Table-type NIT-FSTGEN
  Table-type NIT-FSTPM  for surface layer
  Table-type NIT-FSTPM  for upper layer
  Table-type NIT-FSTPM  for lower layer
  Table-type NIT-FSTPM  for groundwater layer
                                              --
  Table-type NIT-CMAX                         |
  Table-type NIT-SVALPM for surface layer     | if        (single value
  Table-type NIT-SVALPM for upper layer       | FORAFG=    Freundlich
  Table-type NIT-SVALPM for lower layer       | 1          method)
  Table-type NIT-SVALPM for groundwater layer |
                                              --
                                                       --
  Table-type NIT-STOR1  for surface layer storage      | repeat for
  Table-type NIT-STOR1  for upper layer princ. storage | each block
  Table-type NIT-STOR2  for upper layer trans. storage |
                                                       --
  Table-type NIT-STOR1  for lower layer storage
  Table-type NIT-STOR1  for groundwater layer storage
```

Explanation:

The exact format of each of the tables mentioned above, except SOIL-DATA, is detailed in the documentation which follows. SOIL-DATA is documented under the input for Section PEST (4.4(1).10).

The comments given alongside the table names above indicate:

1. Under which circumstances a table is expected
2. Sequencing information. Note that "repeat for each block" means that the bracketed set of tables is repeated for each areal source block in the pervious land-segment (PLS). The first set is for block 1, second for block 2, etc. The number of blocks (NBLKS) was specified in Table-type GEN-INFO (Sect. 4.4(1).1.3.

Note that if all the fields in a table have default values, the table can be omitted from the User's Control Input. Then, the defaults will be adopted.

VNUTFG and FORAFG are the nitrogen plant uptake flag and the ammonium adsorption/desorption method flag respectively. They are described under Table-type NIT-FLAGS (Sect. 4.4(1).11.1) below.

4.4(1).11.1 Table-type NIT-FLAGS -- Flags for nitrogen simulation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-FLAGS
  <-range><------nitflags--------->
  . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . .
  END NIT-FLAGS

*******
Example
*******

  NIT-FLAGS
    <PLS > Nitrogen flags          ***
    # -  # VNUT FORA ITMX BNUM CNUM***
    1    7    1             10   10
  END NIT-FLAGS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<nitflags>	VNUTFG	5I5	0	0	1
	FORAFG		0	0	1
	ITMAXA		30	1	100
	BNUMN		none	1	1000
	CNUMN		none	1	1000

Explanation

If VNUTFG= 1 the first-order plant uptake parameters for nitrogen are allowed to vary throughout the year and four tables of type MON-NITUPT are expected in the User's Control Input. The first appearance is for the surface layer, 2nd for upper layer, 3rd for the lower layer and 4th for the groundwater layer. If VNUTFG=0 the uptake rates do not vary through the year and a value for each layer is specified in a single table (Table-type NIT-UPTAKE).

FORAFG indicates which method is to be used to simulate adsorption and desorption of ammonium:

0 first-order kinetics
1 single-value Freundlich method

ITMAXA is the maximum number of iterations that will be attempted in solving the Freundlich equation; applicable only if FORAFG= 1.

BNUMN is the number of time steps that will elapse between recalculation of biochemical reaction fluxes. For example, if BNUMN= 10 and the simulation time step is 5 min then these fluxes will be recalculated every 50 minutes. All reactions except adsorption/desorption fall into this category. CNUMN is the corresponding number for the chemical (adsorption/desorption) reactions.

4.4(1).11.2 Table-type NIT-UPTAKE -- Nitrogen plant uptake rate parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-UPTAKE
  <-range><---------------uptake----------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END NIT-UPTAKE

*******
Example
*******

  NIT-UPTAKE
    <PLS >Nitrogen plant uptake rates (/day)      ***
    # -  #   Surface     Upper     Lower   Groundw***
    1    2      0.01      0.02      0.01
  END NIT-UPTAKE

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<uptake>	SKPLN,UKPLN, LKPLN,AKPLN	4F10.0	0.0	0.0	none	/day	Both

Explanation

SKPLN, UKPLN, LKPLN and AKPLN are the plant nitrogen uptake reaction rate parameters for the surface, upper, lower and active groundwater layers respectively.

4.4(1).11.3 Table-type MON-NITUPT -- Monthly plant uptake parameters for nitrogen, for the surface, upper, lower or groundwater layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-NITUPT
  <-range><------------------------mon-uptake------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-NITUPT

*******
Example
*******

  MON-NITUPT
    <PLS >  Plant uptake parms for nitrogen in upper layer (/day)    ***
    # - #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    4                 .01  .03  .05  .05  .03  .01
  END MON-NITUPT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-uptake>	KPLNM(*)	12F5.0	0.0	0.0	none	/day	Both

Explanation

This table is required if the plant uptake parameters vary throughout the year (VNUTFG= 1 in Table-type NIT-FLAGS). The entire table is supplied four times; first for the surface layer, second for the upper layer, third for the lower layer and fourth for the active groundwater layer. If omitted, default values will be supplied. For example, if the third and fourth occurrences of the table are omitted, the parameters for the lower and groundwater layers will default to zero.

4.4(1).11.4 Table-type NIT-FSTGEN -- Nitrogen first-order general parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-FSTGEN
  <-range><upt-fact><------------temp-parms--------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END NIT-FSTGEN

*******
Example
*******

  NIT-FSTGEN
    <PLS > Upt-facts<--------- Temp-parms (theta) --------->***
    # -  #  NO3  NH4  PLN KDSA KADA KIMN  KAM KDNI  KNI KIMA***
    1    7   .5   .5 1.07 1.08
  END NIT-FSTGEN

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<upt-fact>	NO3UTF	2F5.0	1.0	0.001	1.0	none	Both
	NH4UTF		0.0	0.0	1.0	none	Both
<temp-parms>	THPLN	8F5.0	1.07	1.0	2.0	none	Both
	THKDSA		1.05	1.0	2.0	none	Both
	THKADA		1.05	1.0	2.0	none	Both
	THKIMN		1.07	1.0	2.0	none	Both
	THKAM		1.07	1.0	2.0	none	Both
	THKDNI		1.07	1.0	2.0	none	Both
	THKNI		1.05	1.0	2.0	none	Both
	THKIMA		1.07	1.0	2.0	none	Both

Explanation

These general parameters apply to nitrogen reactions in all the layers; thus, this table only appears once (or not at all, if defaults are used).

NO3UTF and NH4UTF are parameters intended to designate which fraction of nitrogen uptake comes from nitrate and ammonium, respectively. Their sum should be 1.0

The remaining fields specify the temperature coefficients (theta) for the various reactions:

THPLN	Plant uptake
THKDSA	Ammonium desorption (only relevant if FORAFG= 0)
THKADA	Ammonium adsorption (" " " ")
THKIMN	Nitrate immobilization
THKAM	Organic N ammonification
THKDNI	NO3 denitrification
THKNI	Nitrification
THKIMA	Ammonium immobilization

4.4(1).11.5 Table-type NIT-FSTPM -- Nitrogen first-order reaction parameters for the surface, upper, lower or active groundwater layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-FSTPM
  <-range><----------------------fstparms--------------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END NIT-FSTPM

*******
Example
*******

  NIT-FSTPM
    <PLS >*** Nitrogen first-order parameters for lower layer (/day)
    # -  #***  KDSAM     KADAM     KIMNI       KAM      KDNI        KNI     KIMAM
    1    7       .05       .03                 .02                  .05
  END NIT-FSTPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<fstparms>	KDSAM,KADAM, KIMNI,KAM,KDNI, KNI,KIMAM	7F10.0	0.0	0.0	none	/day	Both

Explanation

These are the first-order reaction rate parameters for a layer of soil:

KDSAM	Ammonium desorption (irrelevant if FORAFG= 1)
KADAM	Ammonium adsorption (" " ")
KIMNI	Nitrate immobilization
KAM	Organic N ammonification
KDNI	Denitrification
KNI	Nitrification
KIMAM	Ammonium immobilization

HSPF expects this table to appear four times in the User's Control Input; first for the surface layer, second for the upper layer, third for the lower layer, fourth for the active groundwater layer. If one or more occurrences of the table are missing, all reaction parameters for the affected layer(s) will be defaulted to zero.

4.4(1).11.6 Table-type NIT-CMAX — Maximum solubility of ammonium

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-CMAX
  <-range><--cmax-->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END NIT-CMAX

*******
Example
*******

  NIT-CMAX
    <PLS >      CMAX***
    # -  #     (ppm)***
    1    5      15.0
  END NIT-CMAX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<cmax>	CMAX	F10.0	0.0	0.0	none	ppm	Both

Explanation

CMAX is the maximum solubility of ammonium in water. This table only appears once and is only required if FORAFG= 1 (that is, adsorption/desorption is simulated using single-value Freundlich method).

4.4(1).11.7 Table-type NIT-SVALPM -- Nitrogen single value Freundlich adsorption/desorption parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-SVALPM
  <-range><-----------svalpm----------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END NIT-SVALPM

*******
Example
*******

  NIT-SVALPM
    <PLS >      XFIX        K1        N1***
    # -  #     (ppm)                    ***
    1    3      10.0       5.0       1.2
  END NIT-SVALPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<svalpm>	XFIX	3F10.0	0.0	0.0	none	ppm	Both
	K1		0.0	0.0	none		Both
	N1		none	1.0	none		Both

Explanation

This table is only required if FORAFG=1; that is, adsorption and desorption of ammonium is simulated using the single value Freundlich method.

This table is exactly analogous to Table-type PEST-SVALPM.

4.4(1).11.8 Table-type NIT-STOR1 — Initial storage of nitrogen in the surface, upper, lower or groundwater layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-STOR1
  <-range><------------------nit-stor1-------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END NIT-STOR1

*******
Example
*******

  NIT-STOR1
    <PLS >Initial storage of N in upper layer (Block1) (lb/ac)***
    # -  #      ORGN      AMAD      AMSU       NO3      PLTN ***
    1    4                          100.      500.       50.
  END NIT-STOR1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<nit-stor1>	ORGN,AMAD,AMSU, NO3,PLTN	5F10.0	0.0 0.0	0.0 0.0	none none	lb/ac kg/ha	Engl Metric

Explanation

This table is similar in organization to Table-type PEST-STOR1. It specifies the initial storage of N in one of the four major soil storages and the plant N derived from that layer. In the case of the surface and upper principal storages the table values refer to a single block, if the PLS is divided into more than one block. The values in the table are:

ORGN	Organic N
AMAD	Adsorbed ammonium
AMSU	Solution ammonium
NO3	Nitrate
PLTN	N stored in plants, derived from this layer (and block, where applicable)

4.4(1).11.9 Table-type NIT-STOR2 — Initial storage of nitrogen in upper layer transitory (interflow) storage

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NIT-STOR2
  <-range><-----nit-stor2---->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END NIT-STOR2

*******
Example
*******

  NIT-STOR2
    <PLS >  Initial N in interflow storage  (Block 3) (lb/ac)***
    #  - #     IAMSU      INO3                              ***
    1    2
  END NIT-STOR2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<nit-stor2>	IAMSU,INO3	2F10.0	0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric

Explanation

This table is similar to Table-type PEST-STOR2. It specifies the initial storage of ammonium and nitrate in the upper layer transitory (interflow) storage, either for the whole PLS (NBLKS=1) or for one block of it (NBLKS> 1).

4.4(1).12 PERLND BLOCK -- Section PHOS input

Layout:

```
  Table-type SOIL-DATA  if sections PEST and NITR are inactive
  Table-type PHOS-FLAGS
  Table-type PHOS-UPTAKE ------------------------ if VPUTFG= 0
                                                --
  Table-type MON-PHOSUPT for surface layer      | if VPUTFG= 1
  Table-type MON-PHOSUPT for upper layer        |
  Table-type MON-PHOSUPT for lower layer        |
  Table-type MON-PHOSUPT for groundwater layer  |
                                                --
  Table-type PHOS-FSTGEN
  Table-type PHOS-FSTPM  for surface layer
  Table-type PHOS-FSTPM  for upper layer
  Table-type PHOS-FSTPM  for lower layer
  Table-type PHOS-FSTPM  for groundwater layer
                                                --
  Table-type PHOS-CMAX                          |
  Table-type PHOS-SVALPM for surface layer      | if        (single value
  Table-type PHOS-SVALPM for upper layer        | FORPFG=    Freundlich
  Table-type PHOS-SVALPM for lower layer        | 1          method)
  Table-type PHOS-SVALPM for groundwater layer  |
                                                --

                                                         --
  Table-type PHOS-STOR1  for surface layer storage       | repeat for
  Table-type PHOS-STOR1  for upper layer princ. storage  | each block
  Table-type PHOS-STOR2  for upper layer trans. storage  |
                                                         --
  Table-type PHOS-STOR1  for lower layer storage
  Table-type PHOS-STOR1  for groundwater layer storage
```

Explanation:

The exact format of each of the tables mentioned above, except SOIL-DATA, is detailed in the documentation which follows. SOIL-DATA is documented under the input for Section PEST (4.4(1).10).

The comments given alongside the table names above indicate:

1. Under which circumstances a table is expected
2. Sequencing information. Note that "repeat for each block" means that the bracketed set of tables is repeated for each areal source block in the pervious land-segment (PLS). The first set is for block 1, second for block 2, etc. The number of blocks (NBLKS) was specified in Table-type GEN-INFO (Sect. 4.4(1).1.3).

Note that if all the fields in a table have default values, the table can be omitted from the User's Control Input. Then, the defaults will be adopted.

VPUTFG and FORPFG are the phosphorus plant uptake flag and the phosphate adsorption/desorption method flag respectively. They are described under Table-type PHOS-FLAGS (Sect. 4.4(1).12.1) below.

4.4(1).12.1 Table-type PHOS-FLAGS -- Flags governing simulation of phosphorus

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-FLAGS
  <-range><-------phosflags------->
  . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . .
  END PHOS-FLAGS

*******
Example
*******

  PHOS-FLAGS
    <PLS > VPUT FORP ITMX BNUM CNUM ***
    # -  #
    1    4    1             10   10
  END PHOS-FLAGS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<phosflags>	VPUTFG	5I5	0	0	1
	FORPFG		0	0	1
	ITMAXP		30	1	100
	BNUMP		none	1	1000
	CNUMP		none	1	1000

Explanation

This table is exactly analogous to Table-type NIT-FLAGS.

4.4(1).12.2 Table-type PHOS-UPTAKE -- Phosphorus plant uptake parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-UPTAKE
  <-range><--------------phos-uptake------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END PHOS-UPTAKE

*******
Example
*******

  PHOS-UPTAKE
    <PLS > Phosphorus plant uptake parms (/day)   ***
    # -  #     SKPLP     UKPLP     LKPLP     AKPLP***
    1           .005       .03       .05       .01
  END PHOS-UPTAKE

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<phos-uptake>	SKPLP, UKPLP, LKPLP, AKPLP	4F10.0	0.0	0.0	none	/day	Both

Explanation

This table is exactly analogous to Table-type NIT-UPTAKE.

4.4(1).12.3 Table-type MON-PHOSUPT — Monthly plant uptake parameters for phosphorus, for the surface, upper, lower or groundwater layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-PHOSUPT
  <-range><-----------------------mon-phosupt----------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-PHOSUPT

*******
Example
*******

  MON-PHOSUPT
    <PLS > Monthly phosphorus uptake parameters for surface layer (/day)***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC
    1    2                 .01  .03  .07  .07  .04  .01
  END MON-PHOSUPT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-phosupt>	KPLPM(*)	12F5.0	0.0	0.0	none	/day	Both

Explanation

This table is exactly analogous to Table-type MON-NITUPT.

4.4(1).12.4 Table-type PHOS-FSTGEN -- Temperature correction parameters for phosphorus reactions

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-FSTGEN
  <-range><-------------------theta------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PHOS-FSTGEN

*******
Example
*******

  PHOS-FSTGEN
    <PLS > Temperature corection parameters (theta)         ***
    # -  #     THPLP    THKDSP    THKADP    THKIMP     THKMP***
    1                     1.07                          1.05
  END PHOS-FSTGEN

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<theta>	THPLP	5F10.0	1.07	1.0	2.0	none	Both
	THKDSP		1.05	1.0	2.0	none	Both
	THKADP		1.05	1.0	2.0	none	Both
	THKIMP		1.07	1.0	2.0	none	Both
	THKMP		1.07	1.0	2.0	none	Both

Explanation

This table is analogous to Table-type NIT-FSTGEN, except for the first two values in that table. The temperature correction parameters supplied in this table (and the reactions they affect) are:

THPLP	Plant uptake
THKDSP	Phosphate desorption (only relevant if FORPFG=0 in Table PHOS-FLAGS)
THKADP	Phosphate adsorption (" " " " " " ")
THKIMP	Phosphate immobilization
THKMP	Organic P mineralization

4.4(1).12.5 Table-type PHOS-FSTPM -- Phosphorus first-order reaction parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-FSTPM
  <-range><---------------phos-fstpm------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END PHOS-FSTPM

*******
Example
*******

  PHOS-FSTPM
    <PLS > Phosphorus first-order parameters for surface layer (/day) ***
    # -  #     KDSP      KADP      KIMP       KMP                     ***
    1    5                                    .04
  END PHOS-FSTPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<phos-fstpm>	KDSP,KADP, KIMP,KMP	4F10.0	0.0	0.0	none	/day	Both

Explanation

This table is analogous to Table-type NIT-FSTPM. The reaction rate parameters supplied in this table are:

KDSP	Phosphate desorption (only used if FORPFG=0 in Table-type PHOS-FLAGS)
KADP	Phosphate adsorption (" " " " " " ")
KIMP	Phosphate immobilization
KMP	Organic P mineralization

4.4(1).12.6 Table-type PHOS-CMAX — Maximum solubility of phosphate

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-CMAX
  <-range><--cmax-->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END PHOS-CMAX

*******
Example
*******

  PHOS-CMAX
    <PLS >      CMAX***
    # -  #     (ppm)***
    1    2       5.0
  END PHOS-CMAX

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<cmax>	CMAX	F10.0	0.0	0.0	none	ppm	Both

Explanation

This table is exactly analogous to Table-type NIT-CMAX.

4.4(1).12.7 Table-type PHOS-SVALPM -- Phosphorus single value Freundlich adsorption/desorption parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-SVALPM
  <-range><---------svalpm------------>
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END PHOS-SVALPM

*******
Example
*******

  PHOS-SVALPM
    <PLS > Parameters for Freundlich method (lower layer) ***
    # -  #      XFIX      K1        N1                     ***
    1            30.     5.0       1.5
  END PHOS-SVALPM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<svalpm>	XFIX	3F10.0	0.0	0.0	none	ppm	Both
	K1		0.0	0.0	none		Both
	N1		none	1.0	none		Both

Explanation

This table is exactly analogous to Table-type NIT-SVALPM. It is only used if FORPFG= 1 in Table-type PHOS-FLAGS.

4.4(1).12.8 Table-type PHOS-STOR1 -- Initial phosphorus storage in the surface, upper, lower or groundwater layer

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-STOR1
  <-range><-------------phos-stor1------------->
  . . . . . . . . . . . . . . . . . . . . . . . . 
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . 
  END PHOS-STOR1

*******
Example
*******

  PHOS-STOR1
    <PLS >Initial phosphorus in upper layer, block2 (lb/ac) ***
    # -  #      ORGP      P4AD      P4SU      PLTP          ***
    1    3       50.     2000.      200.
  END PHOS-STOR1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<phos-stor1>	ORGP,P4AD,	4F10.0	0.0	0.0	none	lb/ac	Engl
	P4SU,PLTP		0.0	0.0	none	kg/ha	Metric

Explanation

This table is analogous to Table-type NIT-STOR1.

4.4(1).12.9 Table-type PHOS-STOR2 -- Initial storage of phosphate in upper layer transitory (interflow) storage

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHOS-STOR2
  <-range><--phos-->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END PHOS-STOR2

*******
Example
*******

  PHOS-STOR2
    <PLS >Phosphate in interflow  in block 3 (kg/ha) ***
    #  -  #     IP4SU                                ***
     1    6      100.
  END PHOS-STOR2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<phos>	IP4SU	F10.0	0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric

Explanation

This table is analogous to Table-type NIT-STOR2.

4.4(1).13 PERLND BLOCK -- Section TRACER input

Layout:

```
  Table-type TRAC-ID
                         --
  Table-type TRAC-TOPSTOR ¦ repeat for each block
                         --
  Table-type TRAC-SUBSTOR
```

Explanation:

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Note that if all the fields in a table have default values, the table can be omitted from the User's Control Input. Then, the defaults will be adopted.

4.4(1).13.1 Table-type TRAC-ID — Name of conservative (tracer) substance

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  TRAC-ID
  <-range><-----trac-id------>
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END TRAC-ID

*******
Example
*******

  TRAC-ID
    <PLS >Name of tracer      ***
    #    #                    ***
    1   10 Chloride
  END TRAC-ID

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<trac-id>	TRACID(*)	5A4	none	none	none

Explanation

Any 20 character string can be supplied as the name of the tracer substance.

4.4(1).13.2 Table-type TRAC-TOPSTOR -- Initial quantity of tracer in topsoil storages

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  TRAC-TOPSTOR
  <-range><---------trac-topstor------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END TRAC-TOPSTOR

*******
Example
*******

  TRAC-TOPSTOR
    <PLS >Initial storage of chloride in topsoil (block 4) (kg/ha) ***
    # -  #     STRSU     UTRSU     ITRSU                           ***
    1                     200.
  END TRAC-TOPSTOR

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<trac-topstor>	STRSU, UTRSU, ITRSU	3F10.0	0.0 0.0	0.0 0.0	none none	lb/ac kg/ha	Engl Metric

Explanation

This table specifies the initial storage of tracer (conservative) in the surface, upper principal and upper transitory storages. If the PLS is subdivided into more than one block, it is repeated for each block.

4.4(1).13.3 Table-type TRAC-SUBSTOR -- Initial quantity of tracer in lower and groundwater storages

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  TRAC-SUBSTOR
  <-range><---trac-substor--->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END TRAC-SUBSTOR

*******
Example
*******

  TRAC-SUBSTOR
    <PLS >Initial storage of chloride in subsoil layers (lb/ac) ***
    # -  #     LTRSU     ATRSU                                  ***
    1           300.      500.
  END TRAC-SUBSTOR

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<trac-substor>	LTRSU,ATRSU	2F10.0	0.0	0.0	none	lb/ac	Engl
			0.0	0.0	none	kg/ha	Metric

Explanation

This table specifies the initial storage of conservative (tracer) material in the lower and active groundwater layers.

4.4(2) IMPLND Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

IMPLND
  General input
 [section ATEMP input]
 [section SNOW input]
 [section IWATER input]
 [section SOLIDS input]
 [section IWTGAS input]
 [section IQUAL input]
END IMPLND

********************************************************************************
```

Explanation

This block contains the data which are "domestic" to all the Impervious Land-segments in the RUN. The "General input" is always relevant: other input is only required if the module section concerned is active.

4.4(2).1 IMPLND BLOCK -- General input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  Table-type ACTIVITY
 [Table-type PRINT-INFO]
  Table-type GEN-INFO

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] above are not always required; for example, because all the values can be defaulted.

4.4(2).1.1 Table-type ACTIVITY -- Active Sections Vector

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ACTIVITY
  <-range><---------a-s-vector--------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END ACTIVITY

*******
Example
*******

  ACTIVITY
    <ILS >               Active Sections  ***
    # -  # ATMP SNOW IWAT  SLD   IWG IQAL ***
    1    7    1    1    1
    9         0    0    0    1
  END ACTIVITY

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<a-s-vector>	ASVEC(6)	6I5	0	0	1

Explanation

The IMPLND module is divided into 6 sections. The values supplied in this table specify which sections are active and which are not, for each operation involving the IMPLND module. A value of 0 means "inactive" and 1 means "active". Any meaningful subset of sections may be active.

4.4(2).1.2 Table-type PRINT-INFO -- Printout information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PRINT-INFO
  <-range><--------print-flags---------><piv><pyr>
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END PRINT-INFO

*******
Example
*******

  PRINT-INFO
    <ILS > ******** Print-flags ******** PIVL  PYR
    # -  # ATMP SNOW IWAT  SLD  IWG IQAL *********
    1    7    2    4    6                  10   12
  END PRINT-INFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<print-flags>	PFLAG(6)	6I5	4	2	6
<piv>	PIVL	I5	1	1	1440
<pyr>	PYREND	I5	9	1	12

Explanation

HSPF permits the user to vary the printout level (maximum frequency) for the various active sections of an operation. The meaning of each permissible value for PFLAG() is:

2 means every PIVL intervals
3 means every day
4 means every month
5 means every year
6 means never

In the example above, output from Impervious Land-segments 1 thru' 7 will occur as follows:

Section	Max frequency
ATEMP	10 intervals
SNOW	month
IWATER	never
SOLIDS thru' IQUAL	month (defaulted)

A value need only be supplied for PIVL if one or more sections have a printout level of 2. For those sections, printout will occur every PIVL intervals (that is, every PDELT=PIVL*DELT mins). PIVL must be chosen such that there are an integer no. of PDELT periods in a day.

HSPF will automatically provide printed output at all standard intervals greater than the specified minimum interval. In the above example, output for section ATEMP will be printed at the end of each 10 intervals, day, month and year.

PYREND is the calendar month which will terminate the year for printout purposes. Thus, the annual summary can reflect the situation over the past water year or the past calendar year, etc.

4.4(2).1.3 Table-type GEN-INFO -- Other general information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  GEN-INFO
  <-range><---ILS-id---------><--unit-syst--><-printu->
  . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END GEN-INFO

*******
Example
*******

  GEN-INFO
    <ILS >      Name             Unit-systems   Printer***
    # -  #                     User  t-series Engl Metr***
                                      in  out          ***
    1      Chicago loop
    2      Astrodome              1    1        23
  END GEN-INFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<ILS-id>	LSID(10)	10A2	none	none	none
<unit-syst>	UUNITS,IUNITS, OUNITS	3I5	1	1	2
<printu>	PUNIT(2)	2I5	0	6	99

Explanation

Any string of up to 20 characters may be supplied as the identifier for an ILS.

The values supplied for <unit-syst> indicate the system of units for data in the UCI, input time series and output time series respectively: 1 means English units, 2 means Metric units.

The values supplied for <printu> indicate the destinations of printout in English and Metric units respectively. A value 0 means no printout is required in that system. A non-zero value means printout is required in that system and and the value is the Fortran unit no. of the file to which the printout is to be written. Note that printout for each Impervious Land Segment can be obtained in either the English or Metric systems, or both (irrespective of the system used to supply the inputs).

4.4(2).2 IMPLND BLOCK -- SECTION ATEMP INPUT

This section, ATEMP, is common to the PERLND and IMPLND modules. See Section 4.4(1).2 for documentation.

4.4(2).3 IMPLND BLOCK -- SECTION SNOW INPUT

This section, SNOW, is common to the PERLND and IMPLND modules. See Section 4.4(1).3 for documentation.

4.4(2).4 IMPLND BLOCK -- Section IWATER input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
 [Table-type IWAT-PARM1 ]
  Table-type IWAT-PARM2
 [Table-type IWAT-PARM3 ]
                            --
 [Table-type MON-RETN   ]     ¦ only reqd if the relevant quantity
 [Table-type MON-MANNING]     ¦ varies through the year
                            --
 [Table-type IWAT-STATE1]
********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [] above are not always required; for example, because all the values can be defaulted.

4.4(2).4.1 Table-type IWAT-PARM1 — First group of IWATER parms (flags)

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWAT-PARM1
  <-range><-------iwatparm1------->
  . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . .
  END IWAT-PARM1

*******
Example
*******

  IWAT-PARM1
    <ILS >          Flags           ***
    # -  # CSNO RTOP  VRS  VNN RTLI ***
    1    7    1    1
  END IWAT-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<iwatparm1>	CSNOFG,RTOPFG, VRSFG,VNNFG, RTLIFG	5I5	0	0	1

Explanation

If CSNOFG is 1, section IWATER assumes that snow accumulation and melt is being considered. It will, therefore, expect that the time series produced by section SNOW are available, either internally (produced in this RUN) or from external sources (produced in a previous RUN). If CSNOFG is 0, no such time series are expected. See the functional description for further information.

If RTOPFG is 1, routing of overland flow is done in exactly the same way as in NPS. A value of 0 results in a new algorithm being used.

The flags beginning with "V" indicate whether or not certain parameters will be assumed to vary through the year: 1 means they do vary, 0 means they do not. The quantities concerned are:

VRSFG	retention storage capacity
VNNFG	Manning's n for the overland flow plane

If either of these flags are on, monthly values for the parameter concerned must be supplied (see Table-types MON- , documented later).

If RTLIFG is 1, any lateral surface inflow to the ILS will be subject to retention storage; if it is 0, it will not.

4.4(2).4.2 Table-type IWAT-PARM2 — Second group of IWATER parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWAT-PARM2
  <-range><-------------iwatparm2----------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END IWAT-PARM2

*******
Example
*******

  IWAT-PARM2
    <ILS >                                                   ***
    # -  #      LSUR      SLSUR      NSUR      RETSC         ***
    1    7      400.       .001
  END IWAT-PARM2

********************************************************************************
Details
```

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<iwatparm2>	LSUR,	F10.0	none none	1.0 0.3	none none	ft m	Engl Metric
	SLSUR,	F10.0	none	.000001	10.	none	Both
	NSUR,	F10.0	0.1	0.001	1.0		Both
	RETSC	F10.0	0.0 0.0	0.0 0.0	10.0 250.	in mm	Engl Metric

Explanation

LSUR is the length of the assumed overland flow plane, and SLSUR is the slope.

NSUR is Manning's n for the overland flow plane.

RETSC is the retention (interception) storage capacity of the surface.

4.4(2).4.3 Table-type IWAT-PARM3 -- Third group of IWATER parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWAT-PARM3
  <-range><----iwatparm3----->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END IWAT-PARM3

*******
Example
*******
  IWAT-PARM3
    <ILS >***
    # -  #*** PETMAX    PETMIN
    1    7
    9             39        33
  END IWAT-PARM3

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<iwatparm3>	PETMAX,	F10.0	40.	none	none	degF	Engl
			4.5	none	none	degC	Metric
	PETMIN	F10.0	35.	none	none	degF	Engl
			1.7	none	none	degC	Metric

Explanation

PETMAX is the air temp below which E-T will arbitrarily be reduced below the value obtained from the input time series, and PETMIN is the temp below which E-T will be zero regardless of the value in the input time series. These values are only used if snow is being considered (CSNOFG= 1).

In the above example, both parameters will be suppplied default values for Land-segments 1 through 7, but the user has over-ridden the defaults for Land-segment 9.

4.4(2).4.4 Table-type MON-RETN -- Monthly retention storage capacity

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-RETN
  <-range><-----------------mon-retn--------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-RETN

*******
Example
*******

  MON-RETN
    <ILS >  Retention storage capacity at start of each month        ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .02  .03  .03  .04  .05  .08  .12  .15  .12  .05  .03  .01

  END MON-RETN

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-retn>	RETSCM(12)	12F5.0	0.0	0.0	10.	in	Engl
			0.0	0.0	250.	mm	Metric

Explanation

Only required if VRSFG in Table-type IWAT-PARM1 is 1.

4.4(2).4.5 Table-type MON-MANNING -- Monthly Manning's n values

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-MANNING
  <-range><----------------mon-Manning------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)            .
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-MANNING

*******
Example
*******

  MON-MANNING
    <ILS >  Manning's n at start of each month                           ***
    # -  # JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC ***
    1    7 .23  .34  .34  .35  .28  .35  .37  .35  .28  .29  .30  .30
  END MON-MANNING

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-Manning>	NSURM(12)	12F5.0	.10	.001	1.0	complex	Both

Explanation

This table is only required if VNNFG in Table-type IWAT-PARM1 is 1.

4.4(2).4.6 Table-type IWAT-STATE1 — IWATER state variables

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWAT-STATE1
  <-range><----iwat-state1--->
                      . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END IWAT-STATE1

*******
Example
*******

  IWAT-STATE1
    <ILS >  IWATER state variables***
    # -  #***    RETS      SURS
    1    7       0.05      0.10
  END IWAT-STATE1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<iwat-state1>	RETS	2F10.0	.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric
	SURS		.001	.001	100	inches	Engl
			.025	.025	2500	mm	Metric

Explanation

This table is used to specify the initial water storages.

RETS is the retention storage.
SURS is the surface (overland flow) storage.

4.4(2).5 IMPLND BLOCK — Section SOLIDS input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type SLD-PARM1]           Tables in brackets [] are
  Table-type SLD-PARM2            not always required.
 [Table-type MON-ACCUM]
 [Table-type MON-REMOV]
 [Table-type SLD-STOR ]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

4.4(2).5.1 Table-type SLD-PARM1 — First group of SOLIDS parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SLD-PARM1
  <-range><--sld-parm1-->
  . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . .
  END SLD-PARM1

*******
Example
*******

  SLD-PARM1
    <ILS >                ***
    # -  # VASD VRSD SDOP***
    1    7    0    1    0
  END SLD-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<sld-parm1>	VASDFG	3I5	0	0	1
	VRSDFG		0	0	1
	SDOPFG		0	0	1

Explanation

If VASDFG is 1, the accumulation rate of solids is allowed to vary throughout the year and Table-type MON-ACCUM is expected. If the flag is zero, the accumulation rate is constant (specified in Table-type SLD-PARM2).The corresponding flag for the unit removal rate is VRSDFG.

If SDOPFG is 1, removal of sediment from the land surface will be simulated with the algorithm used in the NPS model. If it is 0, the new algorithm will be used.

4.4(2).5.2 Table-type SLD-PARM2 — Second group of SOLIDS parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SLD-PARM2
  <-range><-------------sld-parm2--------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . .
  END SLD-PARM2

*******
Example
*******
  SLD-PARM2
    <ILS >***
    # -  #      KEIM      JEIM    ACCSDP    REMSDP***
    1    7      0.08      1.90      0.01       0.5
  END SLD-PARM2
********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sld-parm2>	KEIM	4F10.0	0.0	0.0	none	complex	Both
	JEIM		none	none	none	complex	Both
	ACCSDP		0.0	0.0	none	tons /ac.day	Engl
			0.0	0.0	none	tonnes /ha.day	Metric
	REMSDP		0.0	0.0	1.0	/day	Both

Explanation

KEIM is the coefficient in the solids washoff equation.
JEIM is the exponent in the solids washoff equation.
ACCSDP is the rate at which solids are placed on the land surface.
REMSDP is the fraction of solids storage which is removed each day; when there is no runoff, for example, because of street sweeping.

If monthly values for the accumulation and unit removal rates are being supplied, values supplied for these variables in this table are not relevant.

4.4(2).5.3 Table-type MON-ACCUM -- Monthly solids accumulation rates

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-ACCUM
  <-range><---------------mon-accum----------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-ACCUM

*******
Example
*******

  MON-ACCUM
    <ILS >  Monthly values for solids accumulation (tonnes/ha.day)    ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  0.0  .12  .12  .24  .24  .56  .67  .56  .34  .34  .23  .12
  END MON-ACCUM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-accum>	ACCSDM(12)	12F5.0	0.0	0.0	none	tons/ ac.day	Engl
			0.0	0.0	none	tonnes/ ha.day	Metr

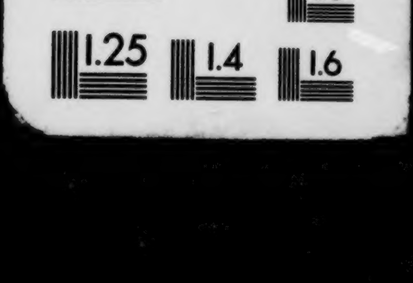
Explanation

This table is only required if VASDFG in Table-type SLD-PARM1 is 1.

6 OF 8

PB80

191109



4.4(2).5.4 Table-type MON-REMOV -- Monthly solids unit removal rates

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-REMOV
  <-range><--------------mon-remov---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-REMOV

*******
Example
*******

  MON-REMOV
    <ILS >  Monthly solids unit removal rate                          ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .05  .05  .07  .15  .15  .20  .20  .20  .20  .10  .05  .05
  END MON-REMOV

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-remov>	REMSDM(12)	12F5.0	0.0	0.0	1.0	/day	Both

Explanation

This table is only required if VRSDFG in Table-type SLD-PARM1 is 1.

4.4(2).5.5 Table-type SLD-STOR -- Solids storage

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SLD-STOR
  <-range><sld-stor>
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END SLD-STOR

*******
Example
*******

  SLD-STOR
    <ILS >  Solids storage (tons/acre) ***
    # -  #                             ***
    1    7       0.2
  END SLD-STOR

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sld-stor>	SLDS	F10.0	0.0	0.0	none	tons/ac	Engl
			0.0	0.0	none	tonnes /ha	Metric

Explanation

SLDS is the initial storage of solids.

4.4(2).6 IMPLND BLOCK -- Section IWTGAS input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type IWT-PARM1]
 [Table-type IWT-PARM2]                   Tables in brackets [] are not
 [Table-type MON-AWTF]                    always required
 [Table-type MON-BWTF]
 [Table-type IWT-INIT]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

4.4(2).6.1 Table-type IWT-PARM1 — Flags for section IWTGAS

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWT-PARM1
  <-range><iwtparm1>
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END IWT-PARM1

*******
Example
*******

  IWT-PARM1
    <ILS >  Flags for section IWTGAS***
    # -  # WTFV CSNO               ***
    1    7    0    0
  END IWT-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<iwtparm1>	WTFVFG	2I5	0	0	1
	CSNOFG		0	0	1

Explanation

WTFVFG indicates whether or not the water temperature regression parameters (AWTF and BWTF) are allowed to vary throughout the year and, thus, whether or not Table-types MON-AWTF and MON-BWTF are expected.

If CSNOFG=1, the effects of snow accumulation and melt are being considered; if it is zero, they are not. If section IWATER is active the value of CSNOFG supplied here is ignored because it was first supplied in the input for that section.

4.4(2).6.2 Table-type IWT-PARM2 — Second group of IWTGAS parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWT-PARM2
  <-range><---------iwt-parm2---------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END IWT-PARM2

*******
Example
*******

  IWT-PARM2
    <ILS >  Second group of IWTGAS parms***
    # -  #      ELEV      AWTF      BWTF***
    1    7     1281.      40.0       0.8
  END IWT-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<iwt-parm2>	ELEV	3F10.0	0.0	-1000.	30000.	ft	Engl
			0.0	-300.	9100.	m	Metric
	AWTF		32.	0.0	100.	DegF	Engl
			0.0	-18.	38.	DegC	Metr
	BWTF		1.0	0.001	2.0	DegF/F	Engl
			1.0	0.001	2.0	DegC/C	Metr

Explanation

ELEV is the elevation of the ILS above sea level (used to adjust saturation concentrations of dissolved gases in surface outflow).

AWTF is the surface water temperature, when the air teperature is 32 degrees F (0 degrees C). It is the intercept of the surface water temperature regression equation. BWTF is the slope of the surface water temperature regression equation.

4.4(2).6.3 Table-type MON-AWTF -- Monthly values for AWTF

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-AWTF
  <-range><-----------------mon-awtf--------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-AWTF

*******
Example
*******

  MON-AWTF
    <ILS >  Value of AWTF at start of each month (deg F)              ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  37.  38.  39.  40.  41.  42.  43.  44.  45.  44.  41.  40.
  END MON-AWTF

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-awtf>	AWTFM(12)	12F5.0	32.	0.	100.	deg F	Engl
			0.	-18.	38.	deg C	Metric

Explanation

This table is only required if WTFVFG in Table-type IWT-PARM1 is 1.

4.4(2).6.4 Table-type MON-BWTF -- Monthly values for BWTF

```
********************************************************************************
          1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-BWTF
  <-range><------------------mon-bwtf--------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-BWTF

*******
Example
*******


  MON-BWTF
    <ILS >  Value of BWTF at start of each month (deg F/F)            ***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7   .3   .3   .3   .4   .4   .5   .5   .5   .4   .4   .4   .3
  END MON-BWTF

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mon-bwtf>	BWTFM(12)	12F5.0	1.0	0.001	2.0	deg F/F	Engl
			1.0	0.001	2.0	deg C/C	Metric

Explanation

This table is only required if WTFVFG in Table-type IWT-PARM1 is 1.

4.4(2).6.5 Table-type IWT-INIT -- Initial conditions for section IWTGAS

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  IWT-INIT
  <-range><---------iwt-init---------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END IWT-INIT

*******
Example
*******

  IWT-INIT
    <ILS >     SOTMP     SODOX     SOCO2***
    # -  #      DegC      mg/l    mg C/l***
    1    7       16.
  END IWT-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<iwt-init>	SOTMP	3F10.0	60.0	32.	100.	Deg F	Engl
			16.0	.01	38.0	Deg C	Metric
	SODOX		0.0	0.0	20.0	mg/l	Both
	SOCO2		0.0	0.0	1.0	mg C/l	Both

Explanation

These are the initial values for the temperature, DO content and CO2 content of the surface runoff. The values given in this table do not affect anything in the simulation beyond the start of the first interval of the run. Therefore, in most runs, this table should be omitted.

4.4(2).7 IMPLND BLOCK -- Section IQUAL input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type NQUALS]
                               --
  Table-type QUAL-PROPS          ¦
 [Table-type QUAL-INPUT]         ¦
 [Table-type MON-POTFW]          ¦ repeat for each
 [Table-type MON-ACCUM]          ¦ quality constituent
 [Table-type MON-SQOLIM]         ¦
                               --
********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows or in the documentation for the PERLND module.

Tables enclosed in brackets [] are not always required;for example, because all the values can be defaulted.

4.4(2).7.1 Table-type NQUALS -- Total number of quality constituents simulated

This table is identical to the corresponding table for the PERLND module. See Section 4.4(1).8.1 for documentation.

4.4(2).7.2 Table-type QUAL-PROPS — Identifiers and Flags for a quality constituent

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  QUAL-PROPS
  <-range><-qualid->      <qt><------flags------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END QUAL-PROPS

*******
Example
*******

  QUAL-PROPS
     <ILS >   Identifiers and Flags               ***
    #  -  #   QUALID      QTID  QSD VPFW  QSO  VQO***
    1     7      BOD        kg     0    0    1    1
  END QUAL-PROPS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<qualid>	QUALID	5A2	none	none	none
<qt>	QTYID	2A2	none	none	none
<flags>	QSDFG	4I5	0	0	1
	VPFWFG		0	0	1
	QSOFG		0	0	1
	VQOFG		0	0	1

Explanation

QUALID is a string of up to 10 characters which identifies the quality constituent. QTYID is a string of up to 4 characters which identifies the units associated with this constituent (e.g., kg, # (for coliforms)). These are the units referred to as "qty" in subsequent tables (eg. Table-type QUAL-INPUT).

If QSDFG is 1 then:

1. This constituent is a QUALSD (sediment associated).
2. If VPFWFG is 1, the washoff potency factor may vary throughout the year. Table-type MON-POTFW is expected.

If QSOFG is 1 then:

1. This constituent is a QUALOF (directly associated with overland flow).
2. If VQOFG is 1 then rate of accumulation and the limiting storage of QUALOF may vary throughout the year. Table-types MON-ACCUM and MON-SQOLIM are expected.

4.4(2).7.3 Table-type QUAL-INPUT -- Storage on surface and nonseasonal parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  QUAL-INPUT
  <-range><------------qual-input---------------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END QUAL-INPUT

*******
Example
*******

  QUAL-INPUT
    <ILS >  Storage on surface and nonseasonal parameters***
    # -  #     SQO   POTFW   ACQOP  SQOLIM   WSQOP        ***
    1    7    1.21    .172    0.02     2.0    1.70
  END QUAL-INPUT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<qual-input>	SQO	5F8.0	0.0	0.0	none	qty/ac	Engl
			0.0	0.0	none	qty/ha	Metric
	POTFW		0.0	0.0	none	qty/ton	Engl
			0.0	0.0	none	qty /tonne	Metric
	ACQOP		0.0	0.0	none	qty /ac.day	Engl
			0.0	0.0	none	qty /ha.day	Metric
	SQOLIM		0.01	0.01	none	qty/ac	Engl
			0.02	0.02	none	qty/ha	Metric
	WSQOP		1.64	0.01	none	in/hr	Engl
			41.7	0.25	none	mm/hr	Metric

Explanation

The following variable is applicable only if the constituent is a QUALSD:

1. POTFW, the washoff potency factor.

A potency factor is the ratio of constituent yield to sediment outflow.

The following variables are applicable only if the constituent is a QUALOF:

1. SQO, the initial storage of QUALOF on the surface of the ILS.
2. ACQOP, the rate of accumulation of QUALOF.
3. SQOLIM, the maximum storage of QUALOF.
4. WSQOP, the rate of surface runoff which will remove 90 percent of stored QUALOF per hour.

If monthly values are being supplied for any of these quantities, the value in this table is not relevant; instead,the system expects and uses values supplied in Table-type MON-XXX.

4.4(2).7.4 Table-type MON-POTFW — Monthly washoff potency factor

This table is identical to the corresponing table in for the PERLND module. See Section 4.4(1).8.4 for documentation.

4.4(2).7.5 Table-type MON-ACCUM — Monthly accumulation rates of QUALOF

This table is identical to the corresponding table for the PERLND module. See Section 4.4(1).8.6 for documentation.

4.4(2).7.6 Table-type MON-SQOLIM — Monthly limiting storage of QUALOF

This table is identical to the corresponding table for the PERLND module. See Section 4.4(1).8.7 for documentation.

4.4(3) RCHRES Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

RCHRES
  General input
 [section HYDR input]
 [section ADCALC input]
 [section CONS input]
 [section HTRCH input]
 [section SED input]
 [section FSTORD input]
 [input for RQUAL sections]
 [section OXRX input]
 [section NUTRX input]
 [section PLANK input]
 [section PHCARB input]
END RCHRES

********************************************************************************
```

Explanation

This block contains the data which are "domestic" to all RCHRES processing units in the RUN. The "General input" is always relevant: other input is only required if the module section concerned is active.

4.4(3).1 RCHRES BLOCK -- General input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  Table-type ACTIVITY
 [Table-type PRINT-INFO]
  Table-type GEN-INFO

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows. Tables enclosed in brackets [], above, are not always required; for example, because all values can be defaulted.

4.4(3).1.1 Table-type ACTIVITY -- Active Sections Vector

```
********************************************************************************
                   1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ACTIVITY
  <-range><-----------------a-s-vector--------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END ACTIVITY

*******
Example
*******

  ACTIVITY
    RCHRES  Active sections***
    # -  # HYFG ADFG CNFG HTFG SDFG FSFG OXFG NUFG PKFG PHFG***
    1    7    1    1    1    1    1    1    1    0    0    0
  END ACTIVITY

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<a-s-vector>	HYDRFG,ADFG CONSFG,HTFG, SEDFG,FSTFG, OXFG,NUTFG, PLKFG,PHFG	10I5	0	0	1

Explanation

The RCHRES module is divided into ten sections. The values supplied in this table specify which sections are active and which are not, for each operation involving the RCHRES module. A value of 0 means "inactive" and 1 means "active". Any meaningful subset of sections may be active, with the following provisos: 1. Section ADCALC must be active if any "quality" sections (CONS thru PHCARB) are active. 2. If any section in the RQUAL group (Section OXRX thru PHCARB) is active, all preceding RQUAL sections must also be active.

4.4(3).1.2 Table-type PRINT-INFO -- Printout information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PRINT-INFO
  <-range><---------------print-flags----------------------><piv><pyr>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PRINT-INFO

*******
Example
*******

  PRINT-INFO
    RCHRES  Printout level flags***
    # -  # HYDR ADCA CONS HEAT  SED  FST OXRX NUTR PLNK PHCB PIVL  PYR***
    1    7    2    2    2    5    5    2    3    3              10   12
  END PRINT-INFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<print-flags>	PFLAG(10)	10I5	4	2	6
<pivl>	PIVL	I5	1	1	1440
<pyr>	PYREND	I5	9	1	12

Explanation

HSPF permits the user to vary the printout level (maximum frequency) for the various active sections of an operation. The meaning of each permissible value for PFLAG( ) is:

2 means every PIVL intervals
3 means every day
4 means every month
5 means every year
6 means never

In the example above, output from RCHRESs 1 through 7 will occur as follows:

Section	Max frequency
HYDR	10 intervals
ADCALC	10 intervals
CONS	10 intervals
HTRCH	year
SED	year
FSTORD	10 intervals
OXRX	day
NUTRX	day
PLANK	month (defaulted)
PHCARB	month (defaulted)

A value need only be supplied for PIVL if one or more sections have a printout level of 2. For those sections, printout will occur every PIVL intervals (that is, every PDELT=PIVL*DELT mins). PIVL must be chosen such that there are an integer no. of PDELT periods in a day.

HSPF will automatically provide printed output at all standard intervals greater than the specified minimum interval. In the above example, output for section NUTRX will be printed at the end of each day, month, and year.

PYREND is the calendar month which will terminate the year for printout purposes. Thus, the annual summary can reflect the situation over the past water year or the past calendar year, etc.

4.4(3).1.3 Table-type GEN-INFO — Other general information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  GEN-INFO
  <-range><-----rchid--------><nex><--unit-syst--><-printu->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END GEN-INFO

*******
Example
*******

  GEN-INFO
    RCHRES       Name        Nexits   Unit Systems     Printer***
    # - #                             user t-series  Engl Metr***
                                           in  out          ***
    4     East River-mile 4       2    1    1        23
  END GEN-INFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<rchid>	RCHID(10)	10A2	none	none	none
<nex>	NEXITS	I5	1	1	5
<unit-syst>	UUNITS	I5	1	1	2
	IUNITS	I5	1	1	2
	OUNITS	I5	1	1	2
<printu>	PUNIT(2)	2I5	0	6	99

Explanation

Any string of up to 20 characters may be supplied as the identifier for a RCHRES. NEXITS is the no. of exits from the RCHRES. A maximum of 5 exits may be handled.

The values supplied for <unit-syst> indicate the system of units for data in the UCI, input time series, and output time series, respectively. 1 means English units, 2 means metric units.

The values supplied for <printer> indicate the destinations of printout in English and metric units, respectively. A value of 0 means no printout is required in that system. A non-zero value means printout is required in that system and is the Fortran unit no. of the file to which printout is to be written.

4.4(3).2 RCHRES BLOCK -- Section HYDR input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  Table-type HYDR-PARM1
  Table-type HYDR-PARM2
 [Table-type MON-CONVF]
 [Table-type HYDR-INIT]

********************************************************************************
```

Explanation

The exact format of each of the tables mentioned above is detailed in the documentation which follows.

Tables enclosed in brackets [], above, are not always required.

4.4(3).2.1 Table-type HYDR-PARM1 — Flags for HYDR section

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  HYDR-PARM1
  <-range> <v><1><2>     <---odfvfg---->     <---odgtfg---->     <----funct---->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END HYDR-PARM1

*******
Example
*******

  HYDR-PARM1
   RCHRES   Flags for HYDR section***
   #  -  #  VC A1 A2***  ODFVFG for each     ODGTFG for each     FUNCT  for each
            FG FG FG***  possible   exit     possible   exit     possible   exit
    1    7   0  1  1       0  0  0  0  1       1  1  1  1  1       3  3  3  3  3
  END HYDR-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<v>	VCONFG	I3	0	0	1
<1>	AUX1FG	I3	0	0	1
<2>	AUX2FG	I3	0	0	1
<odfvfg>	ODFVFG(5)	5I3	0	-5	8
<odgtfg>	ODGTFG(5)	5I3	0	0	5
<funct>	FUNCT(5)	5I3	1	1	3

Explanation

A value of 1 for VCONFG means that F(VOL) outflow demand components are multiplied by a factor which is allowed to vary through the year. These monthly adjustment factors are input in Table-type MON-CONVF in this section.

A value of 1 for AUX1FG means subroutine AUXIL will be called to compute depth, stage, surface area, average depth, and topwidth, and values for these parameters will be reported in the printout. A value of 0 supresses the calculation and printout of this information.

A value of 1 for AUX2FG means average velocity and average cross sectional area will be calculated, and values for these parameters will be reported in the printout. A value of 0 supresses the calculation and printout of this information.

The value specified for ODFVFG determines the F(VOL) component of the outflow demand. A value of 0 means that the outflow demand does not have a volume dependent component. A value greater than 0 indicates the column number in RCHTAB which contains the F(VOL) component. If the value specified for ODFVFG is less than 0, the absolute value indicates the element of array COLIND( ) which defines a pair of columns in RCHTAB which are used to evaluate the F(VOL) component. Further explanation of this latter option is provided in the functional description of the HYDR section in Part E. A value of ODFVFG can be specified for each exit from a RCHRES.

The value specified for ODGTFG determines the G(T) component of the outflow demand. A value of 0 means that the outflow demand does not have such a component. A value greater than 0 indicates the element number of the array OUTDGT( ) which contains the G(T) component. A value of ODGTFG can be specified for each exit from a RCHRES.

FUNCT determines the function used to combine the components of an outflow demand. The possible values and their meanings are:

1 means use the smaller of F(VOL) and G(T)
2 means use the larger of F(VOL) and G(T)
3 means use the sum of F(VOL) and G(T)

4.4(3).2.2 Table-type HYDR-PARM2 — Parameters for HYDR section

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  HYDR-PARM2
  <-range><---------------hydr-parm2---------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END HYDR-PARM2

*******
Example
*******

  HYDR-PARM2
    RCHRES ***
    # - #    FTABNO       LEN     DELTH     STCOR        KS***
    1            17       2.7      120.       3.2        .5
  END HYDR-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<hydr-parm2>	FTABNO	5F10.0	none	1	999	none	Both
	LEN		none none	0.01 0.016	none none	miles km	Engl Metric
	DELTH		0.0 0.0	0.0 0.0	none none	ft m	Engl Metric
	STCOR		0.0 0.0	none none	none none	ft m	Engl Metric
	KS		0.0	0.0	.99	none	Both

Explanation

FTABNO is the user's number for the F-Table which contains the geometric and hydraulic properties of the RCHRES. The F-Table is situated in the F-TABLES block.

LEN is the length of the RCHRES.

DELTH is the drop in water elevation from the upstream to the downstream extremities of the RCHRES. (It is only used if section OXRX is active and reaeration is being computed using the Tsivoglou-Wallace equation.)

STCOR is the correction to the RCHRES depth to calculate stage. (Depth + STCOR = Stage)

KS is the weighting factor for hydraulic routing. Choice of a realistic KS value is discussed in the functional description of the HYDR section in Part E.

4.4(3).2.3 Table-type MON-CONVF -- Monthly f(VOL) adjustment factors

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  MON-CONVF
  <-range><---------------mon-convf------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END MON-CONVF

*******
Example
*******
  MON-CONVF
    RCHRES  Monthly f(VOL) adjustment factors***
    # -  #  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC***
    1    7  .97  .89  .89  .91  .93  .93  .94  .95  .95  .98  .98  .97
  END MON-CONVF
********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<mon-convf>	CONVFM(12)	12F5.0	0.0	0.0	none

4.4(3).2.4 Table-type HYDR-INIT -- Initial conditions for HYDR section

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  HYDR-INIT
  <-range><--vol--->     <-------colind---------->     <-------outdgt---------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END HYDR-INIT

*******
Example
*******

  HYDR-INIT
    RCHRES  Initial conditions for HYDR section***
    # -  #***      VOL    Initial  value  of COLIND     initial  value  of OUTDGT
         ***     ac-ft    for each possible   exit     for each possible   exit
    5            3245.      3.4  4.5  4.5  4.5  4.2      2.1  1.2   .5  1.2  1.8
  END HYDR-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<vol>	VOL	F10.0	0.0	0.0	none	acre-ft	Engl
			0.0	0.0	none	Mm3	Metric
<colind>	COLIND(5)	5F5.0	4.0	4.0	8.0	none	Both
<outdgt>	OUTDGT(5)	5F5.0	0.0	0.0	none	ft3/s	Engl
			0.0	0.0	none	m3/s	Metric

Explanation

VOL is the initial volume of water in the RCHRES.

The value of COLIND( ) for an exit indicates the pair of columns used to evaluate the initial value of the F(VOL) component of outflow demand for the exit.

The array OUTDGT( ) specifies the G(T) component of the initial outflow demand for each exit from the RCHRES.

A non-zero value of COLIND(I) is only meaningful if the outflow from exit I has an f(VOL) component. Similarly, a non-zero value for OUTDGT(I) is only meaningful if the outflow from exit I has a g(t) component.

4.4(3).3 RCHRES BLOCK -- Section ADCALC input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type ADCALC-DATA]

********************************************************************************
```

Explanation

The exact format of this input is detailed below. Table ADCALC-DATA is not always required because its contents can be defaulted.

4.4(3).3.1 Table-type ADCALC-DATA -- Data for section ADCALC

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ADCALC-DATA
  <-range><---adcalc-data---->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END ADCALC-DATA

*******
Example
*******

  ADCALC-DATA
    RCHRES  Data for section ADCALC***
    # - #     CRRAT       VOL***
    5           1.7     324.
  END ADCALC-DATA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<adcalc-data>	CRRAT	2F10.0	1.5	1.0	none	none	Both
	VOL		0.0	0.0	none	acre-ft	Engl
			0.0	0.0	none	Mm3	Metric

Explanation

CRRAT is the ratio of maximum velocity to mean velocity in the RCHRES cross section under typical flow conditions.

VOL is the volume of water in the RCHRES at the start of the simulation. Input of this value is not necessary if section HYDR is active.

4.4(3).4 RCHRES BLOCK -- Section CONS input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type NCONS]
                         --
  Table-type CONS-DATA     |   repeat for each conservative constituent
                         --

********************************************************************************
```

Explanation

The exact formats of these tables are detailed below. Table-type NCONS is not required if only one conservative constituent is being simulated (default value).

4.4(3).4.1 Table-type NCONS -- Number of conservative constituents simulated

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NCONS
  <-range><ncn>
  . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . .
  END NCONS

*******
Example
*******

  NCONS
    RCHRES     ***
    # -  #NCONS***
    1    7    4
  END NCONS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<ncn>	NCONS	I5	1	1	10

4.4(3).4.2 Table-type CONS-DATA -- Information about one conservative substance

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
  CONS-DATA
  <-range><----conid---------><---con-->  <concid><--conv-->  <qtyid->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END CONS-DATA
```

****
Example
****

```
CONS-DATA
  RCHRES  Data for conservative constituent No. 3***
  # -  #        Substance-id      Conc    ID      CONV        QTYID***
  1    7  Total Diss  Solids     251.3  mg/l    1000.       kg
END CONS-DATA
```

****

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<conid>	CONID(5)	5A4	none	none	none	none	Both
<con>	CON	F10.0	0.0	0.0	none	concid	Both
<concid>	CONCID	2A4	none	none	none	none	Both
<conv>	CONV	F10.0	none	0.0	none	see below	
<qtyid>	QTYID	2A4	none	none	none	none	Both

Explanation

Any string of up to 20 characters may be supplied as the name of the conservative constituent (CONID).

CON is the initial concentration of the conservative.

CONCID is a string of up to 8 characters which specifies the concentration units for the conservative constituent.

CONV is the conversion factor from QTYID/VOL to the desired concentration units (CONCID): CONC = CONV*(QTY/VOL). If UUNITS is 1, VOL is in ft3; if it is 2, VOL is in m3. For example, if:

CONCID is mg/l
QTYID is kg
VOL is in m3,

then CONV=1000.

QTYID is a string of up to 8 characters which specifies the units in which the total flow of constituent into, or out of, the RCHRES will be expressed, eg "kg".

4.4(3).5 RCHRES BLOCK — Section HTRCH input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type HEAT-PARM]
 [Table-type HEAT-INIT]

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

4.4(3).5.1 Table-type HEAT-PARM -- Parameters for section HTRCH

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  HEAT-PARM
  <-range><--elev--><--eldat-><--cfsx--><--ktrd--><--kcnd--><--kevp-->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END HEAT-PARM

*******
Example
*******
  HEAT-PARM
    RCHRES      ELEV     ELDAT    CFSAEX    KATRAD     KCOND     KEVAP ***
    # -  #        ft        ft                                         ***
    1    7     2000.     1500.        .5       6.5       11.        4.
  END HEAT-PARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<elev>	ELEV	F10.0	0.0	0.0	30000.	ft	Engl
			0.0	0.0	10000.	m	Metric
<eldat>	ELDAT	F10.0	0.0	none	none	ft	Engl
			0.0	none	none	m	Metric
<cfsx>	CFSAEX	F10.0	1.0	0.001	2.0	none	Both
<ktrd>	KATRAD	F10.0	9.37	1.00	20.	none	Both
<kcnd>	KCOND	F10.0	6.12	1.00	20.	none	Both
<kevp>	KEVAP	F10.0	2.24	1.00	10.	none	Both

Explanation

ELEV is the mean RCHRES elevation
ELDAT is the difference in elevation between the RCHRES and the air temperature gage (positive if RCHRES is higher than the gage).
CFSAEX is the correction factor for solar radiation (it includes fraction of RCHRES surface exposed to radiation).
KATRAD is the longwave radiation coefficient
KCOND is the conduction-convection heat transport coefficient.
KEVAP is the evaporation coefficient.

4.4(3).5.2 Table-type HEAT-INIT -- Initial conditions

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  HEAT-INIT
  <-range><----init-temp----->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END HEAT-INIT

*******
Example
*******
  HEAT-INIT
    RCHRES        TW    AIRTMP ***
    # -  #      degF      degF ***
    1    7       62.       70.
  END HEAT-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<init-temp>	TW	F10.0	60.	32.	200.	degF	Engl
			15.5	0.0	95.	degC	Metric
	AIRTMP	F10.0	60.	-90.	150.	degF	Engl
			15.5	-70.0	65.	degC	Metric

Explanation

TW is the water temperature and AIRTMP indicates the air temperature at the RCHRES.

4.4(3).6 RCHRES-BLOCK — Section SED input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  Table-type SED-PARM
 [Table-type SED-INIT]

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

4.4(3).6.1 Table-type SED-PARM — Sediment parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SED-PARM
  <-range><---------sed-parm----------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END SED-PARM

*******
Example
*******

  SED-PARM
    RCHRES    WSHSET     KSAND    EXPSND***
    # - #      ft/hr                    ***
    1    7       .60       20.       3.0
  END SED-PARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sed-parm>	WSHSET	F10.0	0.0	0.0	10.	ft/hr	Engl
			0.0	0.0	3.	m/hr	Metric
	KSAND	F10.0	none	0.001	none		Both
	EXPSND	F10.0	none	0.01	none		Both

Explanation

WSHSET is the sinking rate of washload material.
KSAND is the sandload suspension coefficient.
EXPSND is the exponent in the sandload suspension equation.

4.4(3).6.2 Table-type SED-INIT -- Initial sediment conditions

```
********************************************************************************
          1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SED-INIT
  <-range><--------sed-init----------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END SED-INIT

*******
Example
*******

  SED-INIT
    RCHRES       WASH      SAND     BDSAND***
    # -  #       mg/l      mg/l       tons***
    1    7      1000.     1200.        62.
  END SED-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<sed-init>	WASH	F10.0	0.0	0.0	none	mg/l	Both
	SAND	F10.0	0.0	0.0	none	mg/l	Both
	BDSAND	F10.0	0.0	none	none	tons	Engl
			0.0	none	none	tonnes	Metric

Explanation

WASH=washload concentration, SAND=sandload concentration and BDSAND=bed storage of sand.

4.4(3).7 RCHRES-BLOCK — Section FSTORD input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type NFSTORD]
                         --
  Table-type FSTOR-DATA   ¦   repeat for each first order constituent
                         --

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

4.4(3).7.1 Table-type NFSTORD -- Number of first order constituents

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NFSTORD
  <-range><nfs>
  . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . .
  END NFSTORD

*******
Example
*******

  NFSTORD
   RCHRES NFSTO***
    # -  #     ***
    1    7    6
  END NFSTORD

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<nfs>	NFSTOR	I5	1	1	10	none	Both

4.4(3).7.2 Table-type FSTOR-DATA -- Data for a first order constituent

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  FSTOR-DATA
  <-range><------fstid-------><-fstor-->  <concid><--conv-->  <qtyid-><kfs><tcf>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END FSTOR-DATA

*******
Example
*******
  FSTOR-DATA
    RCHRES  ***          FSTID     FSTOR    CONCID        CONV      QTYID  KFS TCF
    # -  #  ***                                                            /hr
    1    7           Coliforms       2.0       #/1        .001          #  .02 1.1
  END FSTOR-DATA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<fstid>	FSTID	5A4	none	none	none	none	Both
<fstor>	FSTOR	F10.0	0.0	0.0	none	concid	Both
<concid>	CONCID	2A4	none	none	none	none	Both
<conv>	CONV	F .0.0	none	0.0	none	see below	
<qtyid>	QTYID	2A4	none	none	none	none	Both
<kfs>	KFST20	F5.0	0.0	0.0	100	/hr	Both
<tcf>	TCFST	F5.0	1.07	1.0	2.0	none	Both

Explanation

FSTID - Name of decay constituent
FSTOR - Initial decay constituent concentration
CONCID - Concentration units
CONV - Factor to convert from Qty/Vol to concentration units:
       Conc= Conv* Qty/Vol   (in English system, Vol is in ft3)
       (in Metric system, Vol is in m3)
QTYID - Name of "Qty" unit for decay constituent
KFST20 - First order decay coefficient @ 20 degrees C
TCFST - Temperature correction to decay rate

4.4(3).8 RCHRES-BLOCK — Input for RQUAL sections

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type BENTH-FLAG]
 [Table-type SCOUR-PARMS]
  Section OXRX input
 [Section NUTRX input]   if NUTRX is active
 [Section PLANK input]  if PLANK is active
 [Section PHCARB input] if PHCARB is active

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

4.4(3).8.01 Table-type BENTH-FLAG — Benthic release flag

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  BENTH-FLAG
  <-range><ben>
  . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . .
  END BENTH-FLAG

*******
Example
*******
  BENTH-FLAG
    RCHRES BENF***
    # -  #     ***
    1    7
  END BENTH-FLAG

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ben>	BENRFG	I5	0	0	1	none	Both

Explanation

If BENRFG is 1, benthal influences are considered

4.4(3).8.02 Table-type SCOUR-PARMS -- Benthal scour parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SCOUR-PARMS
  <-range><----scour-parms--->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END SCOUR-PARMS

*******
Example
*******

  SCOUR-PARMS
    RCHRES    SCRVEL    SCRMUL***
    # -  #    ft/sec          ***
    1    7       15.        3.
  END SCOUR-PARMS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<scour-parms>	SCRVEL	F10.0	10.	.01	none	ft/sec	Engl
			3.05	.01	none	m/sec	Metric
	SCRMUL	F10.0	2.0	1.0	none		Both

Explanation

SCRVEL - The velocity above which effects of scouring on benthal release rates is considered.

SCRMUL - Multiplier to increase benthal releases during scouring.

4.4(3).8.1 RCHRES-BLOCK — Section OXRX input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type OX-FLAGS]
  Table-type OX-GENPARM
 [Table-type ELEV]         if section HTRCH is not active
 [Table-type OX-BENPARM]   if BENRFG=1 (Table-type BENTH-FLAG)
 [Table-type OX-CFOREA]    if LKFG=1 (Table-type OX-FLAGS)
                                                                -----
                                                   --- if           ¦
 [Table-type OX-TSIVOGLOU]                           ¦ REAMFG=1     ¦
  Table-type OX-LEN-DELTH  if section HYDR inactive  ¦ (Tsivoglou)  ¦ if
                                                   ---              ¦ LKFG=0
 [Table-type OX-TCGINV]    if REAMFG=2 (Owen/Churchill,etc.)        ¦
  Table-type OX-REAPARM    if REAMFG=3                              ¦
                                                                -----
 [Table-type OX-INIT]

********************************************************************************
```

Explanation

The conditions under which data from the various tables are needed are indicated above. REAMFG is the reaeration method flag, defined in Section 4.4(3).8.1.1 below.

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

4.4(3).8.1.1 Table-type OX-FLAGS — Oxygen flags

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-FLAGS
  <-range><ox-flags>
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END OX-FLAGS

*******
Example
*******

  OX-FLAGS
    RCHRES REAM LKFG***
    # -  #          ***
    1    7    2    0
  END OX-FLAGS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-flags>	REAMFG	I5	2	1	3	none	Both
	LKFG	I5	0	0	1	none	Both

Explanation

REAMFG indicates the method used to calculate reaeration coefficient for free-flowing streams.

1 Means Tsivoglou method is used
2 Means Owens, Churchill, or O'Connor-Dobbins method is used depending on velocity and depth of water
3 Means coefficient is calculated as a power function of velocity and/or depth; user inputs exponents for velocity and depth and an empirical constant (REAK)

If LKFG is 1, the RCHRES is a single layer lake or mixed reservoir; if 0, it is a free-flowing stream.

4.4(3).8.1.2 Table-type OX-GENPARM -- General oxygen parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-GENPARM
  <-range><-------------ox-genparm--------------->
  . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . .
  END OX-GENPARM

*******
Example
*******

  OX-GENPARM
    RCHRES    KBOD20     TCBOD    KODSET    SUPSAT***
    # -  #       /hr               ft/hr          ***
    1    7       0.1      1.06       8.0       1.2
  END OX-GENPARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-genparm>	KBOD20	F10.0	none	0.0	none	/hr	Both
	TCBOD	F10.0	1.075	1.0	2.0	none	Both
	KODSET	F10.0	0.0	0.0	none	ft/hr	Engl
			0.0	0.0	none	m/hr	Metric
	SUPSAT	F10.0	1.15	1.0	2.0	none	Both

Explanation

KBOD20 - Unit BOD decay rate @ 20 degrees C
TCBOD - Temperature correction coefficient for BOD decay
KODSET - Rate of BOD settling
SUPSAT - Allowable dissolved oxygen supersaturation (expressed as a multiple of DO saturation concentration)

4.4(3).8.1.3 Table-type ELEV -- RCHRES elevation above sea level

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ELEV
  <-range><--elev-->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END ELEV

*******
Example
*******

  ELEV
    RCHRES      ELEV***
    # -  #        ft***
    1    7     2100.
  END ELEV

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<elev>	ELEV	F10.0	0.0	0.0	30000	ft	Engl
			0.0	0.0	10000	m	Metric

4.4(3).8.1.4 Table-type OX-BENPARM -- Oxygen benthic parameters

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-BENPARM
  <-range><--------ox-benparm---------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END OX-BENPARM

*******
Example
*******

  OX-BENPARM
    RCHRES     BENOD  BRBOD(1)  BRBOD(2)***
    # -  #mg/c.m2.hr  mg/m2.hr  mg/m2.hr***
    1    7        1.0
  END OX-BENPARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-benparm>	BENOD	F10.0	0.0	0.0	none	mg/DegC.m2.hr	Both
	BRBOD(1)	F10.0	72.	.01	none	mg/m2.hr	Both
	BRBOD(2)	F10.0	100.	.01	none	mg/m2.hr	Both

Explanation

BENOD - Benthal oxygen demand per degree C (with unlimited DO concentration) (demand is, thus, proportional to the water temperature)
BRBOD(1) - Benthal release of BOD at high oxygen concentration.
BRBOD(2) - Increment to benthal release of BOD under anaerobic conditions

4.4(3).8.1.5 Table-type OX-CFOREA -- Lake reaeration correction coefficient

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-CFOREA
  <-range><-cforea->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END OX-CFOREA

*******
Example
*******

  OX-CFOREA
    RCHRES    CFOREA***
    # -  #          ***
    1    7       0.8
  END OX-CFOREA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<cforea>	CFOREA	F10.0	1.0	.001	10.

Explanation

CFOREA is a correction factor in the lake reaeration equation, to account for good or poor circulation characteristics.

4.4(3).8.1.6 Table-type OX-TSIVOGLOU -- Parms for Tsivoglou calculation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-TSIVOGLOU
  <-range><---ox-tsivoglou--->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END OX-TSIVOGLOU

*******
Example
*******

  OX-TSIVOGLOU
    RCHRES      REAKT    TCGINV***
     # -  #       /ft          ***
     1    7       .07       1.1
  END OX-TSIVOGLOU

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-tsivoglou>	REAKT	F10.0	0.08	0.001	1.0	/ft	Both
	TCGINV	F10.0	1.047	1.0	2.0	none	Both

Explanation

REAKT is the empirical constant in Tsivoglou's equation for reaeration (escape coefficient)
TCGINV is the temperature correction coefficient for surface gas invasion.

4.4(3).8.1.7 Table-type OX-LEN-DELTH -- Length of reach and fall

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-LEN-DELTH
  <-range><---ox-len-delth--->
  . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . .
  END OX-LEN-DELTH

*******
Example
*******

  OX-LEN-DELTH
    RCHRES       LEN      DELTH***
    # -  #     miles         ft***
    1    7       10.       200.
  END OX-LEN-DELTH

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-len-delth>	LEN	F10.0	none	.01	none	miles	Engl
			none	.01	none	km	Metric
	DELTH	F10.0	none	0.0	none	ft	Engl
			none	0.0	none	m	Metric

Explanation

LEN is the length of the RCHRES and DELTH is the (energy) drop over its length.

4.4(3).8.1.8 Table-type OX-TCGINV -- Owen/Churchill/O'Connor-Dobbins data (temperature correction coefficient)

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-TCGINV
  <-range><-tcginv->
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END OX-TCGINV

*******
Example
*******

  OX-TCGINV
    RCHRES    TCGINV***
    # -  #          ***
    1    7      1.07
  END OX-TCGINV

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<tcginv>	TCGINV	F10.0	1.047	1.0	2.0

Explanation

TCGNIV is the temperature correction coefficient for surface gas invasion.

4.4(3).8.1.9 Table-type OX-REAPARM -- Parms for user-supplied reaeration formula

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-REAPARM
  <-range><-------------ox-reaparm---------------->
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  (repeats until all operations of this type are covered)
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  END OX-REAPARM

*******
Example
*******

  OX-REAPARM
    RCHRES    TCGINV      REAK    EXPRED    EXPREV***
    # -  #                 /hr                   ***
    1    7      1.08       1.0      -2.0       0.7
  END OX-REAPARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-reaparm>	TCGINV	F10.0	1.047	1.0	2.0	none	Both
	REAK	F10.0	none	0.0	none	/hr	Both
	EXPRED	F10.0	0.0	none	0.0	none	Both
	EXPREV	F10.0	0.0	0.0	none	none	Both

Explanation

TCGINV - See section 4.4(3).8.1.6

REAK - Empirical constant for equation used to calculate reaeration coefficient

EXPRED - Exponent to depth used in calculation of reaeration coefficient.

EXPREV - Exponent to velocity used in calculation of reaeration coefficient

4.4(3).8.1.10 Table-type OX-INIT -- Initial concentrations

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OX-INIT
  <-range><---------ox-init----------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END OX-INIT

*******
Example
*******

  OX-INIT
    RCHRES      DOX       BOD      SATDO***
    # -  #     mg/l      mg/l       mg/l***
    1    7      26.       17.2        43.
  END OX-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ox-init>	DOX	F10.0	0.0	0.0	20.0	mg/l	Both
	BOD	F10.0	0.0	0.0	none	mg/l	Both
	SATDO	F10.0	10.0	0.1	20.0	mg/l	Both

Explanation

DOX - Dissolved oxygen
BOD - Biochemical oxygen demand
SATDO - Dissolved oxygen saturation concentration

4.4(3).8.2 RCHRES-BLOCK -- Section NUTRX input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type NUT-FLAGS]
 [Table-type CONV-VAL1]
 [Table-type NUT-BENPARM]  if BENRFG=1 (Table-type BENTH-FLAG)
  Table-type NUT-NITRIF    if NH3FG=1 (Table-type NUT-FLAGS)
 [Table-type NUT-INIT]

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

BENRFG indicates whether or not benthal influences are considered.
NH3FG indicates whether or not ammonia is simulated.

4.4(3).8.2.1 Table-type NUT-FLAGS -- Nutrient flags

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NUT-FLAGS
  <-range><---------nut-flags---------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END NUT-FLAGS

*******
Example
*******

  NUT-FLAGS
    RCHRES  NH3  NO2  PO4  AMV  DEN DENR***
    # -  #                              ***
    1    7    1         1
  END NUT-FLAGS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<nut-flags>	NH3FG,NO2FG, PO4FG,AMVFG, DENFG,DENRFG	6I5	0	0	1

Explanation

NH3FG - If on, ammonia is simulated
NO2FG - If on, nitrite is simulted
PO4FG - If on, ortho-phosphorus is simulated
AMVFG - If on, ammonia vaporization is enabled
DENFG - If on, denitrification is enabled
DENRFG - If on, denitrification end product is ammonia;
         otherwise end product is nitrogen gas

4.4(3).8.2.2 Table-type CONV-VAL1 -- Conversion factors

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  CONV-VAL1
  <-range><------------conv-val1----------------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END CONV-VAL1

*******
Example
*******

  CONV-VAL1
    RCHRES      CVBO     CVBPC     CVBPN    BPCNTC***
    # - #      mg/mg  mols/mol  mols/mol          ***
    1    7       4.0       67.       33.       77.
  END CONV-VAL1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<conv-val1>	CVBO	F10.0	1.98	1.0	5.0	mg/mg	Both
	CVBPC	F10.0	106.	50.	200.	mols/mol	Both
	CVBPN	F10.0	16.	10.	50.	mols/mol	Both
	BPCNTC	F10.0	49.	10.	100.	none	Both

Explanation

CVBO - Conversion from milligrams biomass to milligrams oxygen

CVBPC - Conversion from biomass expressed as phosphorus to carbon equivalency

CVBPN - Conversion from biomass expressed as phosphorus to nitrogen equivalency

BPCNTC - Percentage, by weight, of biomass which is carbon

4.4(3).8.2.3 Table-type NUT-BENPARM -- Nutrient benthic parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NUT-BENPARM
  <-range><---------------nut-benparm----------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END NUT-BENPARM

*******
Example
*******

  NUT-BENPARM
    RCHRES  BRNIT(1)  BRNIT(2)  BRPO4(1)  BRPO4(2)     ANAER***
    # -  #  mg/m2.hr  mg/m2.hr  mg/m2.hr  mg/m2.hr      mg/l***
    1    7       10.       20.       1.0       4.0      .001
  END NUT-BENPARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<nut-benparm>	BRNIT(1)	5F10.0	11.	.01	none	mg/m2.hr	Both
	BRNIT(2)		33.	.01	none	mg/m2.hr	Both
	BRPO4(1)		1.1	.01	none	mg/m2.hr	Both
	BRPO4(2)		2.2	.01	none	mg/m2.hr	Both
	ANAER		.0005	.0001	1.0	mg/l	Both

Explanation

BRNIT - Benthal release of inorganic nitrogen. (1) indicates aerobic rate and (2) indicates anaerobic rate.
BRPO4 - Benthal release of ortho-phosphate. Subscripts same as BRNIT.
ANAER - Concentration of dissolved oxygen below which anaerobic conditions exist

4.4(3).8.2.4 Table-type NUT-NITRIF -- Nitrification parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NUT-NITRIF
  <-range><--------nut-nitrif---------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END NUT-NITRIF

*******
Example
*******

  NUT-NITRIF
    RCHRES    KNH320    KNO220     TCNIT***
    # -  #       /hr       /hr          ***
    1    7       .05       .05       1.1
  END NUT-NITRIF

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units
<nut-nitrif>	KNH320	3F10.0	none	0.001	none	/hr
	KNO220		none	0.001	none	/hr
	TCNIT		1.2	1.0	2.0	

Explanation

KNH320 and KNO220 are the unit oxidation rates of ammonia and nitrite, respectively, at 20 degrees C.

4.4(3).8.2.5 Table-type NUT-INIT -- Nutrient initial conditions

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  NUT-INIT
  <-range><-----------------nut-init----------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END NUT-INIT

*******
Example
*******

  NUT-INIT
    RCHRES       NO3       NH3       NO2       PO4     DEBAC***
    # -  #      mg/l      mg/l      mg/l      mg/l          ***
    1    7        7.       13.       41.       22.        .6
  END NUT-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<nut-init>	NO3	F10.0	0.0	0.0	none	mg/l	Both
	NH3	F10.0	0.0	0.0	none	mg/l	Both
	NO2	F10.0	0.0	0.0	none	mg/l	Both
	PO4	F10.0	0.0	0.0	none	mg/l	Both
	DEBAC	F10.0	0.0	0.0	1.0	none	Both

Explanation

NO3 - Nitrate (as nitrogen)
NH3 - Ammonia (as nitrogen)
NO2 - Nitrite (as nitrogen)
PO4 - Ortho-phosphorus (as phosphorus)
DEBAC - Dentrifying bacteria

4.4(3).8.3 RCHRES-BLOCK -- Section PLANK input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  Table-type PLNK-FLAGS
  Table-type SURF-EXPOSED     if section HTRCH inactive
  Table-type PLNK-PARM1
 [Table-type PLNK-PARM2]
 [Table-type PLNK-PARM3]
                                                -----
  Table-type PHYTO-PARM                             |
                         ---                        |
  Table-type ZOO-PARM1     | if                     | if
 [Table-type ZOO-PARM2]    | ZOOFG=1                | PHYFG=1
                         ---                        |
                                                -----

 [Table-type BENAL-PARM]  if BALFG=1
 [Table-type PLNK-INIT]

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

PHYFG, ZOOFG and BALFG are flags which indicate whether or not phytoplankton, zooplankton and benthic algae are being simulated. They are documented under Table-type PLNK-FLAGS below.

4.4(3).8.3.1 Table-type PLNK-FLAGS -- Plankton flags

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
  PLNK-FLAGS
  <-range><-------------plnk-flags--------------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END PLNK-FLAGS
```

****
Example
****

```
  PLNK-FLAGS
    RCHRES PHYF ZOOF BALF SDLT AMRF DECF NSFG ZFOO***
    # -  #                                        ***
    1    7    1         1                        3
  END PLNK-FLAGS
```

****

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<plnk-flags>	PHYFG,ZOOFG, BALFG,SDLTFG, AMRFG,DECFG, NSFG	7I5	0	0	1
	ZFOOD	I5	2	1	3

Explanation

The following, except for ZFOOD, are the conditions when the flag is on:

PHYFG - Phytoplankton is simulated
ZOOFG - Zooplankton are simulated
BALFG - Benthic algae are simulated
SDLTFG - Influence of sediment washload on light extinction is simulated
AMRFG - Ammonia retardation of nitrogen limited growth is enabled
DECFG - Linkage between carbon dioxide and phytoplankton growth is decoupled
NSFG - Ammonia is included as part of available nitrogen supply in nitrogen limited growth calculations

ZFOOD - The quality of zooplankton food

4.4(3).8.3.2 Table-type SURF-EXPOSED -- Correction factor for solar radiation data

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SURF-EXPOSED
  <-range><surf-exp>
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END SURF-EXPOSED

*******
Example
*******

  SURF-EXPOSED
    RCHRES    CFSAEX***
    # -  #          ***
    1    7        .5
  END SURF-EXPOSED

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<surf-exp>	CFSAEX	F10.0	0.0	0.0	1.0	none	Both

Explanation

This factor is used to adjust the input solar radiation to make it applicable to the RCHRES; for example, to account for shading of the surface by trees or buildings.

4.4(3).8.3.3 Table-type PLNK-PARM1 -- General plankton parms, group 1

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PLNK-PARM1
  <-range><--------------------plnk-parm1-------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PLNK-PARM1

*******
Example
*******

  PLNK-PARM1
    RCHRES    RATCLP    NONREF    LITWSH     ALNPR      EXTB     MALGR***
    # -  #                                              /ft        /hr***
    1    7        .5        .3                  .4     0.1
  END PLNK-PARM1

********************************************************************************
Details
```

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<plnk-parm1>	RATCLP	F10.0	.6	.01	none	none	Both
	NONREF	F10.0	.5	.01	1.0	none	Both
	LITWSH	F10.0	0.0	0.0	none	1/mg.ft	Both
	ALNPR	F10.0	1.0	.01	1.0	none	Both
	EXTB	F10.0	none	.001	none	/ft	Engl
			none	.001	none	/m	Metric
	MALGR	F10.0	.3	.001	none	/hr	Both

Explanation

RATCLP - Ratio of chlorophyll "A" content of biomass to phosphorus content
NONREF - Nonrefractory fraction of algae and zooplankton biomass
LITWSH - Multiplication factor to washload concentration to determine washload contribution to light extinction
ALNPR - Fraction of nitrogen requirements for phytoplankton growth satisfied by nitrate
EXTB - Base extinction coefficient for light
MALGR - Maximal unit algal growth rate

4.4(3).8.3.4 Table-type PLNK-PARM2 -- General plankton parms, group 2

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PLNK-PARM2
  <-range><--------------------------plnk-parm2-------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PLNK-PARM2

*******
Example
*******

  PLNK-PARM2
    RCHRES *** CMMLT      CMMN     CMMNP      CMMP    TALGRH    TALGRL    TALGRM
    # -  # ***ly/min      mg/l      mg/l      mg/l      degF      degF      degF
    1    7       .01       .05       .04                85.0      44.0      71.0
  END PLNK-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<plnk-parm2>	CMMLT	F10.0	.033	1.0E-6	none	ly/min	Both
	CMMN	F10.0	.045	1.0E-6	none	mg/l	Both
	CMMNP	F10.0	.0284	1.0E-6	none	mg/l	Both
	CMMP	F10.0	.0150	1.0E-6	none	mg/l	Both
	TALGRH	F10.0	95.	50.	212.	degF	Engl
			35.	10.	100.	degC	Metric
	TALGRL	F10.0	43.	32.	212.	degF	Engl
			6.1	0.0	100.	degC	Metric
	TALGRM	F10.0	77.	32.	212.	degF	Engl
			25.	0.0	100.	degC	Metric

Explanation

CMMLT - Michaelis-Menten constant for light limited growth
CMMN - Nitrate Michaelis-Menten constant for nitrogen limited growth
CMMNP - Nitrate Michaelis-Menten constant for phosphorus limited growth
CMMP - Phosphate Michaelis-Menten constant for phosphorus limited growth
TALGRH - Temperature above which algal growth ceases
TALGRL - Temperature below which algal growth ceases
TALGRM - Temperature below which algal growth is retarded

4.4(3).8.3.5 Table-type PLNK-PARM3 -- General plankton parms, group 3

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PLNK-PARM3
  <-range><---------------------plnk-parm3---------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PLNK-PARM3

*******
Example
*******

  PLNK-PARM3
    RCHRES     ALR20     ALDH      ALDL     OXALD     NALDH     PALDH***
    #  -  #      /hr      /hr       /hr       /hr      mg/l      mg/l***
    1     7               .02                 .04
  END PLNK-PARM3

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<plnk-parm3>	ALR20	F10.0	.0002	1.0E-6	none	/hr	Both
	ALDH	F10.0	.01	1.0E-6	none	/hr	Both
	ALDL	F10.0	.001	1.0E-6	none	/hr	Both
	OXALD	F10.0	.03	1.0E-6	none	/hr	Both
	NALDH	F10.0	0.0	0.0	none	mg/l	Both
	PALDH	F10.0	0.0	0.0	none	mg/l	Both

Explanation

ALR20 - Algal unit respiration rate at 20 degrees C
ALDH - High algal unit death rate
ALDL - Low algal unit death rate
OXALD - Increment to phytoplankton unit death rate due to anaerobic conditions
NALDH - Inorganic nitrogen concentration below which high algal death rate occurs (as nitrogen)
PALDH - Inorganic phosphorus concentration below which high algal death rate occurs (as phosphorus)

4.4(3).8.3.6 Table-type PHYTO-PARM -- Phytoplankton parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PHYTO-PARM
  <-range><----------------------phyto-parm-------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PHYTO-PARM

*******
Example
*******
  PHYTO-PARM
    RCHRES      SEED    MXSTAY      OREF    CLALDH    PHYSET    REFSET***
    # -  #      mg/l      mg/l     ft3/s      ug/l     ft/hr     ft/hr***
    1    7       2.0       15.       8.0
  END PHYTO-PARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<phyto-parm>	SEED	F10.0	0.0	0.0	none	mg/l	Both
	MXSTAY	F10.0	0.0	0.0	none	mg/l	Both
	OREF	F10.0	0.0	0.0	none	ft3/s	Engl
			0.0	0.0	none	m3/s	Metric
	CLALDH	F10.0	50.0	.01	none	ug/l	Both
	PHYSET	F10.0	0.0	0.0	none	ft/hr	Engl
			0.0	0.0	none	m/hr	Metric
	REFSET	F10.0	0.0	0.0	none	ft/hr	Engl
			0.0	0.0	none	m/hr	Metric

Explanation

SEED - Minimum concentration of plankton not subject to advection (i.e. at high flow).
MXSTAY - Concentration of plankton not subject to advection at very low flow
OREF - Outflow at which concentration of plankton not subject to advection is midway between SEED and MXSTAY
CLALDH - Chlorophyll "A" concentration above which high algal death rate occurs
PHYSET - Rate of phytoplankton settling
REFSET - Rate of settling for dead refractory organics

4.4(3).8.3.7 Table-type ZOO-PARM1 -- First group of zooplankton parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ZOO-PARM1
  <-range><------------------zoo-parm1---------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END ZOO-PARM1

*******
Example
*******

  ZOO-PARM1
    RCHRES    MZOEAT    ZFIL20    ZRES20        ZD      OXZD***
    # -  #  mg/l.hr l/mgzoo.hr       /hr       /hr       /hr***
    1   7       .098       0.2
  END ZOO-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<zoo-parm1>	MZOEAT	F10.0	.055	.001	none	mg phyto/ mg zoo.hr	Both
	ZFIL20	F10.0	none	0.001	none	l/mgzoo.hr	Both
	ZRES20	F10.0	.0015	1.0E-6	none	/hr	Both
	ZD	F10.0	.0001	1.0E-6	none	/hr	Both
	OXZD	F10.0	.03	1.0E-6	none	/hr	Both

Explanation

MZOEAT - Maximum zooplankton unit ingestion rate
ZFIL20 - Zooplankton filtering rate at 20 degrees C
ZRES20 - Zooplankton unit respiration rate at 20 degrees C
ZD - Natural zooplankton unit death rate
OXZD - Increment to unit zooplankton death rate due to anaerobic conditions

4.4(3).8.3.8 Table-type ZOO-PARM2 -- Second group of zooplankton parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  ZOO-PARM2
  <-range><--------------zoo-parm2--------------->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END ZOO-PARM2

*******
Example
*******

  ZOO-PARM2
    RCHRES    TCZFIL    TCZRES    ZEXDEL    ZOMASS***
    # -  #                                  mg/org***
    1    7       1.2       1.1       0.8
  END ZOO-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<zoo-parm2>	TCZFIL	F10.0	1.17	1.0	2.0	none	Both
	TCZRES	F10.0	1.07	1.0	2.0	none	Both
	ZEXDEL	F10.0	0.7	.001	1.0	none	Both
	ZOMASS	F10.0	.0003	1.0E-6	1.0	mg/org	Both

Explanation

TCZFIL and TCZRES are the temperature correction coefficients for filtering and respiration, respectively.
ZEXDEL is the fraction of nonrefractory zooplankton excretion which is immediately decomposed when ingestion rate > MZOEAT.
ZOMASS is the average weight of a zooplankton organism.

4.4(3).8.3.9 Table-type BENAL-PARM -- Benthic algae parms

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  BENAL-PARM
  <-range><---------benal-parm--------->
  . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . .
  END BENAL-PARM

*******
Example
*******

  BENAL-PARM
    RCHRES      MBAL    CFBALR    CFBALG***
    # -  #     mg/m2                    ***
    1    7      520.       .56       .80
  END BENAL-PARM

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<benal-parm>	MBAL	F10.0	600.	.01	none	mg/m2	Both
	CFBALR	F10.0	1.0	.01	1.0	none	Both
	CFBALG	F10.0	1.0	.01	1.0	none	Both

Explanation

MBAL is the maximum benthic algae density (as biomass)
CFBALR and CFBALG are the ratios of benthic algal to phytoplankton respiration and growth rates, respectively.

4.4(3).8.3.10 Table-type PLNK-INIT -- Initial plankton conditions

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PLNK-INIT
  <-range><----------------------plank-init-------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END PLNK-INIT

*******
Example
*******

  PLNK-INIT
    RCHRES     PHYTO       ZOO     BENAL       ORN       ORP       ORC***
   #  -  #      mg/l     org/l     mg/m2      mg/l      mg/l      mg/l***
    1    7     .0001       .05      .002       .01       .02       .01
  END PLNK-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<plank-init>	PHYTO	F10.0	.96E-6	1.0E-10	none	mg/l	Both
	ZOO	F10.0	.03	1.0E-6	none	org/l	Both
	BENAL	F10.0	1.0E-8	1.0E-10	none	mg/m2	Both
	ORN	F10.0	0.0	0.0	none	mg/l	Both
	ORP	F10.0	0.0	0.0	none	mg/l	Both
	ORC	F10.0	0.0	0.0	none	mg/l	Both

Explanation

PHYTO - Phytoplankton, as biomass
ZOO - Zooplankton
BENAL - Benthic algae, as biomass
ORN - Dead refractory organic nitrogen
ORP - Dead refractory organic phosphorus
ORC - Dead refractory organic carbon

4.4(3).8.4 RCHRES-BLOCK -- Section PHCARB input

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

 [Table-type PH-PARM1]
 [Table-type PH-PARM2]
 [Table-type PH-INIT ]

********************************************************************************
```

Explanation

The exact format of each of the tables above is detailed in the documentation which follows. Tables in brackets [] need not always be supplied; for example, because all of the inputs have default values.

4.4(3).8.4.1 Table-type PH-PARM1 -- Flags for pH simulation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PH-PARM1
  <-range><ph-parm1>
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END PH-PARM1

*******
Example
*******

  PH-PARM1
    RCHRES PHCN ALKC***
    # -  #          ***
    1    7   30    9
  END PH-PARM1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<ph-parm1>	PHCNT	I5	25	1	100
	ALKCON	I5	1	1	10

Explanation

PHCNT - Maximum number of iterations to pH solution
ALKCON - Number of the conservative substance which is alkalinity

4.4(3).8.4.2 Table-type PH-PARM2 -- Parameters for pH simulation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PH-PARM2
  <-range><----------ph-parm2---------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END PH-PARM2

*******
Example
*******

  PH-PARM2
    RCHRES    CFCINV  BRCO2(1)  BRCO2(2)***
    # -  #            mg/m2.hr  mg/m2.hr***
    1    7      .901      72.0      65.1
  END PH-PARM2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ph-parm2>	CFCINV	F10.0	.913	.001	1.0	none	Both
	BRCO2(1)	F10.0	62.	.01	none	mg/m2.hr	Both
	BRCO2(2)	F10.0	62.	.01	none	mg/m2.hr	Both

Explanation

CFCINV - Ratio of carbon dioxide invasion rate to oxygen reaeration rate

BRCO2 - Benthal release of CO2 (as carbon) for (1) aerobic and (2) anaerobic conditions

4.4(3).8.4.3 Table-type PH-INIT -- Initial conditions for pH simulation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  PH-INIT
  <-range><----------ph-init---------->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END PH-INIT

*******
Example
*******

  PH-INIT
    RCHRES       TIC       CO2         PH***
    # -  #      mg/l      mg/l           ***
    1    7       2.0       .03        8.0
  END PH-INIT

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ph-init>	TIC	F10.0	0.0	0.0	none	mg/l	Both
	CO2	F10.0	0.0	0.0	none	mg/l	Both
	PH	F10.0	7.0	1.0	15.0	none	Both

Explanation

TIC - Total inorganic carbon
CO2 - Carbon dioxide (as carbon)
PH - pH

4.4(11) COPY Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

COPY
  Table-type TIMESERIES
END COPY

********************************************************************************
```

Explanation

The COPY module is used to copy one or more time series from one location (source) to another (target). See Section 4.2(11) in Part E for a detailed description of its function.

4.4(11).1 Table-type TIMESERIES -- Number of time series to be copied

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  TIMESERIES
  <-range><npt><nmn>
  . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . .
  END TIMESERIES

*******
Example
*******

  TIMESERIES
  Copy-opn ***
    # - #  NPT  NMN***
    1    7    4
  END TIMESERIES

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<npt>	NPT	I5	0	0	20
<nmn>	NMN	I5	0	0	20

Explanation

NPT is the number of point-valued time series to be copied

NMN is the number of mean-valued time series to be copied

4.4(12) PLTGEN Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
PLTGEN
  Table-type PLOTINFO
  Table-type GEN-LABELS
  Table-type SCALING
  Table-type CURV-DATA  (repeats for each curve to be plotted)
END PLTGEN
********************************************************************************
```

Explanation

The PLTGEN module prepares one or more time series for display on a plotter. It writes the time series, and associated title and scaling information, to a "plot-file" which must be input to a stand-alone program that translates the data into commands that drive the plotter. See Section 4.2(12) of Part E for further details.

4.4(12).1 Table-type PLOTINFO -- General plot information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
  PLOTINFO
  <-range><fil><npt><nmn><lab><pyr>
  . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . .
  END PLOTINFO
*******
Example
*******

  PLOTINFO
  Plot-opn  ***
   # -  # FILE  NPT  NMN LABL PYR ***
    1    3             2
  END PLOTINFO

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<fil>	PLOTFL	I5	30	30	99
<npt>	NPT	I5	0	0	10
<nmn>	NMN	I5	0	0	10
<lab>	LABLFG	I5	0	-1	1
<pyr>	PYREND	I5	9	1	12

Explanation

PLOTFL is the Fortran unit number of the plot file (output of this operation).

NPT is the number of point-valued time series to be plotted.

NMN is the number of mean-valued time series to be plotted.

LABLFG indicates how the plot will be labeled:

> -1 means no labels (useful if you only want to observe the curves, and not have to wait for plotter to add labels).
>
> 0 means standard labeling; that is, one set of X and Y axes and associated labels will be drawn for entire plot.
>
> 1 means separate X and Y axes and labels will be drawn for each "frame" of the plot (eg. each water year). Useful if a long plot is to be reproduced on several successive pages of a report.

PYREND is the calendar month which terminates a plot frame (eg. a water year).

4.4(12).2 Table-type GEN-LABELS -- General plot labels

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  GEN-LABELS
  <-range><--------------- title ---------------->          <------ylabl------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END GEN-LABELS

*******
Example
*******

  GEN-LABELS
  Plot-opn ***
    # - #   General title                                   Y-axis label ***
    1    3  Reservoir inflow and outflow rates              Flow (ft3/sec)
  END GEN-LABELS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<title>	TITLE	20A2	none	none	none
<ylabl>	YLABL	10A2	none	none	none

Explanation

TITLE is the general plot title.

YLABL is the label to be placed on the Y-axis.

4.4(12).3 Table-type SCALING -- Scaling information

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  SCALING
  <-range><--ymin--><--ymax--><--ivlin->
  . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . .
  END SCALING

*******
Example
*******

  SCALING
  Plot-opn ***
    # -  #       YMIN      YMAX     IVLIN ***
    1    3                 500.       48.
  END SCALING

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<ymin>	YMIN	F10.0	0.0	none	none	Note1	Both
<ymax>	YMAX	F10.0	none	none	none	Note1	Both
<ivlin>	IVLIN	F10.0	none	0.01	none	ivl/in	Both

Note1: Units are defined by the user, in field YLABL of Table-type GEN-LABELS

Explanation

YMIN is the minimum ordinate (Y axis) value.

YMAX is the maximum ordinate value.

IVLIN is the horizontal (time) scale; that is, number of intervals (in plot file) per inch on graph.

4.4(12).4 Table-type CURV-DATA -- Data for each curve on plot

Repeats for each curve on the plot

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  CURV-DATA
  <-range>    <-----label---><lin><int><col>
  . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . .
  END CURV-DATA

*******
Example
*******

  CURV-DATA
  Plot-opn    Curve label     Line Intg Color ***
    # -  #                    type equiv code ***
    1    3    Inflow
  END CURV-DATA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<label>	LABEL	8A2	none	none	none
<lin>	LINTYP	I5	0	none	none
<int>	INTEQ	I5	0	0	13
<col>	COLCOD	I5	1	1	4

Explanation

LABEL is the label (descriptor) for this particular curve.

LINTYP describes the type of line to be drawn for this curve. It also determines the frequency of plotted symbols:

> A zero value means points are connected by straight lines; no symbols are drawn at individual data points.
>
> A positive value means points are connected by straight lines; the magnitude determines the frequency of plotted symbols (eg. 4 means plot a symbol at every 4th point obtained from the plot file).
>
> A negative value means no connecting lines are drawn. Only symbols are plotted; the absolute value determines the frequency (as above).

INTEQ is the "integer equivalent" of the symbols to be plotted for this curve (ie. indicates which symbol to use). It is only meaningful if LINTYP is not zero. Value of 2 might mean a triangle, etc.

COLCOD is the color code for this curve. The meaning depends on how the stand-alone plot program is set up; eg. 1 might mean red pen, 2 blue pen, etc.

Note: These data are designed with the requirements of a Calcomp plotting system in mind, but are also useful on some other plotting systems. The stand-alone program, which reads the plot file and drives the plotter, must translate these data into plotter commands.

4.4(13) DISPLY Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

DISPLY
  Table-type DISPLY-INFO1
 [Table-type DISPLY-INFO2]
END DISPLY

********************************************************************************
```

Explanation

The DISPLY module summarizes a time series and presents the results in neatly formatted tables. Data can be displayed at any HSPF-supported interval. See Section 4.2(13) of Part E for further information.

4.4(13).1 Table-type DISPLY-INFO1 -- Contains most of the information necessary to generate data displays.

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  DISPLY-INFO1
  <-range><----------title-------------> <tr><piv>    d<fil><pyr>    d<fil><ynd>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END DISPLY-INFO1

*******
Example
*******

  DISPLY-INFO1
    #thru#***<----------Title---------->      <-short-span->
          ***                                 <---disply--->  <annual summary ->
          ***                            TRAN PIVL DIG1 FIL1  PYR DIG2 FIL2 YRND
    1     Daily precip in TSS #20 (in)                           1    2   20    6
    2     Simulated soil temp (Deg C)    AVER    4    1   21    1    1   22    6
  END DISPLY-INFO1

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<title>	TITLE(*)	15A2	none	none	none
<tr>	TRAN(*)	2A2	SUM	none	none
<piv>	PIVL	I5	0	0	1440
d	DIGIT1 DIGIT2	A1	0	0	7

<fil>	FILE1 FILE2	I5	6	6	99
<pyr>	PYRFG	I5	0	0	1
<ynd>	PYREND	I5	9	1	12

Explanation

TITLE is the title that will be printed at the top of each page of the display.

TRAN is the "transformation code", used to aggregate data from the basic interval (internal time step) to the various display intervals (for both short- and long-span displays). Valid values are: SUM, AVER, MAX, MIN, LAST.

PIVL is the no. of basic time intervals (DELT mins each) to be aggregated to get to the interval of the data printed in a short- span display (eg. In the above example, if DELT were 15 mins for DISPLY operation #2, then the data in the short-span summary tables would be displayed at an interval of 1 hour (PIVL=4). If PIVL=0, a short-span display is not produced.

DIGIT1 and DIGIT2 are the no. of decimal digits to be used to print data in the short-span and long-span displays, respectively. Note that it is up to the user to ensure that this value falls in the valid range 0-7. HSPF does not check this.

FILE1 and FILE2 are the Fortran unit nos. of the files to which short- and long-span displays will be routed.

PYRFG indicates whether or not a long-span display (annual summary of daily values) is required. Value 1 means it is, 0 means it is not.

PYREND is the calendar month which will appear at the right-hand extremity of an annual summary. This enables the user to decide whether the data should be displayed on a calendar year or some other (eg. water year) basis.

4.4(13).2 Table-type DISPLY-INFO2 -- Additional optional information for module DISPLY.

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
  DISPLY-INFO2
  <-range><--mult--><---add--><-thresh1><-thresh2>
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END DISPLY-INFO2

*******
Example
*******

  DISPLY-INFO2
    #thru# Convert DegC to F    Display negative data ***
                Mult       Add    THRSH1              ***
    2    5       1.8      32.0     -999.
  END DISPLY-INFO2

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max	Units	Unit system
<mult>	A	F10.0	1.0	none	none	none	Both
<add>	B	F10.0	0.0	none	none	none	Both
<thresh1>	THRESH1	F10.0	0.0	none	none	none	Both
<thresh2>	THRESH2	F10.0	0.0	none	none	none	Both

Explanation

This table is usually not supplied.

A and B are parameters used to convert the data from internal units to display units:

Display value= A*(internal value)+B

The conversion is done before any aggregation of data to coarser time steps (than DELT) is performed. Note that the default values of A and B result in no change.

THRSH1 and THRSH2 are "threshold values" for the short-span and long-span displays, respectively (THRSH2 is not presently used). THRSH1 can be used to reduce the quantity of printout produced in a short-span display; it functions as follows: When the individual values in a row of the display have been aggregated to get the "row value" (hour- or day-value, depending on the display interval), if the row-value is greater than THRSH1 the row is printed, else it is omitted. Thus, for example, the default of 0.0 will ensure that rows of data containing all zeros are omitted.

4.4(14) DURANL Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******


DURANL
  Table-type GEN-DURDATA
 [Table-type SEASON]
 [Table-type DURATIONS]
 [Table-type LEVELS]
END DURANL

********************************************************************************
```

Explanation

The DURANL module performs duration and excursion analysis on a time series. For example, it analyzes the frequency with which N consecutive values in the time series exceed a specified set of values, called "levels". N is the "duration" of the excursion; up to 10 durations may be used in one duration analysis operation. The user may specify that only those data falling within a specified time in each year (analysis season) be processed. For further details see Section 4.2(14) of Part E.

4.4(14).1 Table-type GEN-DURDATA -- General information for duration analysis

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  GEN-DURDATA
  <-range><----------------title-----------------><-nd><-nl><-pr><-pu>
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END GEN-DURDATA

*******
Example
*******

  GEN-DURDATA
    #thru#<###--------------title----------------> NDUR NLEV PRFG   P-
          ###                                                     UNIT
    1     Simulated DO in Reach 40                           5    2
  END GEN-DURDATA

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<title>	TITLE(*)	20A2	none	none	none
<nd>	NDUR	I5	1	1	10
<nl>	NLEV	I5	1	1	20
<pr>	PRFG	I5	1	1	6
<pu>	PUNIT	I5	6	6	99

Explanation

TITLE is the title which the user gives to the duration analysis operation; usually, something which identifies the time series being analyzed.

NDUR is the no. of durations for which the time series will be analyzed.

NLEV is the no. of "levels" which will be used in analyzing the time series.

PRFG is a flag which governs the quantity of information printed out. A value of 1 results in minimal (basic) output. Increasing the value (up to the maximum of 6) results in increased detail of output

PUNIT is the Fortran unit no. to which the (printed) output of the duration analysis operation will be routed.

4.4(14).2 Table-type SEASON -- The analysis season

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
  SEASON
  <-range>        <---start-->        <----end--->
  . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . .
  END SEASON
*******
Example
*******

  SEASON
          ***      Start               End
    #thru#***      mo da hr mn         mo da hr mn
    1   10         02                  02
  END SEASON

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<start>	SESONS(2-5)	4(1X,I2)	see below		
<end>	SESONE(2-5)	4(1X,I2)	see below		

Explanation

This table is used if one wishes to specify an "analysis season"; that is, that only data falling between the specified starting and ending month/day/ hour/minute (in each year) should be considered.

Note:

1. The defaults, minima, maxima and other values for specifying the starting and ending date/times are the same as those given in the discussion of the GLOBAL Block (Section 4.2). Basically, if any fields in the starting date/time are blank they default to the earliest meaningful value; for the ending date/time they default to the latest possible values. Thus, the analysis season in the example above includes the entire month of February.
2. Although it is not meaningful to provide for a "year" in the fields documented above (since the analysis season applies to every year in the run), the four spaces preceding both the <start> and <end> fields should be left blank because the system does, in fact, read the year and expects it to be blank or zero.
3. The defaults imply that, if this table is omitted, the analysis season extends from January through December.

4.4(14).3 Table-type DURATIONS -- Durations to be used in the analysis

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  DURATIONS
  <-range><-d1><----------------others---------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END DURATIONS

*******
Example
*******

  DURATIONS
    #thru#***<---Durations-------->
          ***  1    2    3    4    5
    1    2     1   10   15   20   40
    3          1   20   21   22
  END DURATIONS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<d1>	DURAT(1)	I5	1	1	1
<others>	DURAT(2-10)	9I5	2	2	none

Explanation

DURAT(*) is an array which contains the NDUR different durations for which the time series will be analyzed (NDUR was specified in Table-type GEN- DURDATA). The durations are expressed in multiples of the internal time step specified in the OPN SEQUENCE Block (Section 4.3). Thus, if DELT= 5 min and the duration is 3, the time series will be analyzed with a "window" of 15 minutes. The analysis algorithm requires that the first duration be 1 time step, but the others can have any value.

4.4(14).4 Table-type LEVELS -- Levels to be used in the analysis

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  LEVELS
  <-range><------------------------- first 7---------------------------------->
  <-range><-------------------------second 7---------------------------------->
  <-range><------------------------- last 6 ------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END LEVELS

*******
Example
*******

  LEVELS
    #thru#***         1         2         3         4         5         6         7
    #thru#***         8         9        10
    1              -30.      -10.        0.       10.       20.       40.       80.
    1              100.      200.     1000.
    #thru#***         1         2         3         4
                   -20.        0.       20.       50.
  END LEVELS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<first7>	LEVEL(2-8)	7F10.0	0.0	none	none
<second7>	LEVEL(9-15)	7F10.0	0.0	none	none
<last6>	LEVEL(16-21)	6F10.0	0.0	none	none

Explanation

LEVEL(2thru21) contains the 20 possible "levels" for which the input time series will be analyzed. (LEVEL(1) and LEVEL(22) are reserved for system use and this does not affect the user since he can only specify LEVEL(2thru21)). The actual no. of levels (NLEV) was specified by the user in Table-type GEN-DURDATA. If NLEV is greater than 7 the entry for a given operation must be continued to the next line; up to 3 lines may be required to cover all the levels. In the example above, operation 1 has 10 levels and thus requires 2 lines, but operation 2 only requires 1 line because it has only 4 levels.

When an entry has to be continued onto more than 1 line:

1. No blank or "comment" lines may be put between any of the lines for a continued entry. Put all comments ahead of the entry. (See operation 1 in above example).
2. The <range> specification must be repeated for each line onto which the entry is continued.

Note that the levels must be specified in ascending order. The system checks that this requirement is not violated.

4.4(15). GENER Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

GENER
  Table-type OPCODE
                       -
 [Table-type NTERMS]    |   only required if
  Table-type COEFFS     |     OPCODE=8
                       -
END GENER

********************************************************************************
```

Explanation

The GENER module generates a time series from one or two input time series. Usually, only Table-type OPCODE is required. However, if OPCODE=8 (power series), you need to supply the no. of terms in the power series and the values of the coefficients.

4.4(15).1 Table-type OPCODE -- Operation code for time series generation

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  OPCODE
  <-range><opn>
  . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . .
  END OPCODE

*******
Example
*******

  OPCODE
    #thru#  OP- ***
           CODE ***
    1    3    8
    5        20
  END OPCODE

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<opn>	OPCODE	I5	none	1	21

Explanation

OPCODE is the operation code. If A and B are the input time series and C is the generated time series, the functions performed for the allowable range of values of OPCODE are:

OPCODE	Definition
1	C= Abs(A)
2	C= Sqrt(A)
3	C= Trunc(A)
4	C= Ceil(A)
5	C= Floor(A)
6	C= loge(A)
7	C= log10(A)
8	C= K(1)+K(2)*A+K(3)*A**2+(up to 7 terms)
9 thru 15	Spare, for future use
16	C= A+B
17	C= A-B
18	C= A*B
19	C= A/B
20	C= Max (A,B)
21	C= Min (A,B)

If 1<=OPCODE<=8, only one input time series is required; else two inputs are required. Note that the operation is performed on the data when they are in internal form (timestep=DELT,etc). For further details, see Section 4.2(15) of Part E.

4.4(15).2 Table-type NTERMS -- No. of terms in power series

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******
  NTERMS
  <-range><-nt>
  . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . .
  END NTERMS

*******
Example
*******

  NTERMS
    #thru#NTERMS ***
    1    2    4
  END NTERMS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<nt>	NTERMS	I5	2	1	7

Explanation

This table is only relevant if OPCODE=8. NTERMS is the total no. of terms in the power series:

C= K(1)+K(2)*A+K(3)*A**2 etc.

The default value of 2 was chosen because this option will probably be used most often (to perform a linear transformation).

4.4(15).3 Table-type COEFFS -- Coefficients in generating power function

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  COEFFS
  <-range><----------------------------coeffs---------------------------------->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all operations of this type are covered)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  END COEFFS

*******
Example
*******

  COEFFS
    #thru# ***    K1        K2        K3
    1    7      -2.0        1.5      0.2
  END COEFFS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<coeffs>	K(*)	7F10.0	0.0	none	none

Explanation

This table is only relevant if OPCODE=8. K(1 thru NTERMS) are the coefficients in the power function:

```
C= K(1)+K(2)*A+K(3)*A**2+ etc.
```

4.5 FTABLES Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

FTABLES
  FTABLE    <t>
<----ftab-parms---->
<------------- row-of-values -------------------------------------------------->
...........................................................................
line above repeats until function has been described through desired range
...........................................................................
  END FTABLE<t>

  FTABLE    <t>
<----ftab-parms---->
<------------- row-of-values -------------------------------------------------->
...........................................................................
line above repeats until function has been described through desired range
...........................................................................
  END FTABLE<t>

  Any number of FTABLES may appear in the block

END FTABLES
********************************************************************************
```

Details

Symbol	FORTRAN Name(s)	Format	Comment
<t>	NUMBR	I3	Users identifying no. for this FTABLE.
<ftab-parms>	Fparms(4)	4I5	Up to 4 control parameters may be supplied for an Ftable, e.g. no. of rows, no. of cols., etc. Exact details will depend on the FTABLE concerned.
<row-of-values>	VAL(*)	variable	Each column is dedicated to one of the variables in the function. Each row contains a full set of corresponding values of these variables. e.g. depth, surface area,volume,outflow for a RCHRES

Explanation

An FTABLE is used to specify, in discrete form, a functional relationship between two or more variables. For example, in the RCHRES module, it is assumed that there is a fixed relationship between depth, surface area, volume, and f(VOL) discharge component. An FTABLE is used to document this nonanalytic function in numerical form. Each column of the FTABLE is dedicated to one of the above variables, and each row contains corresponding values of the set. That is, each row contains the surface area, volume, and discharge for a given depth. The number of rows in the FTABLE will depend on the range of depth to be covered and the desired resolution of the function.

4.5(3) FTABLES for the RCHRES Application Module

4.5(3).1 FTABLE for HYDR section

The geometric and hydraulic properties of a RCHRES are summarised in a function table (FTABLE). Every RCHRES must be associated with one FTABLE; the association is done in Table-type HYDR-PARM2 (Section 4.4(3).2.2 above). Usually, every RCHRES will have its own FTABLE; however, if RCHRESs are identical they can share the same FTABLE.

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

  FTABLE    <t>
<-nr><-nc>
<-depth--><--area--><-volume-><-------------- f(VOL)-values ------------------->
.........................................................................
The above row repeats until values have been supplied to cover the entire
cross section at the desired resolution
.........................................................................
  END FTABLE<t>

*******
Example
*******

  FTABLE    103
 rows cols                                         ***
    3    5
     depth      area    volume  outflow1  outflow2 ***
      (ft)   (acres) (acre-ft)  ( ft3/s)  ( ft3/s) ***
       0.0       0.0       0.0       0.0       0.0
       5.0      10.0      25.0      20.5      10.2
      20.0     120.0    1000.0     995.0     200.1
  END FTABLE103
```

Details

Symbol	FORTRAN Name(s)	Format	Comment
<t>	see Sect. 4.5		
<nr>	NROWS	I5	No. of rows used to document function
<nc>	NCOLS	I5	No. of columns in FTABLE
<depth>	Depth	F10.0	Units: ft or m
<area>	Surface area	F10.0	Units: acres or ha
<volume>	Volume	F10.0	There must be at least one entry with volume =0.0 Units: acre.ft or Mm3
<f(VOL)- values	f(V)(NCOLS-3)	(NCOLS-3) F10.0	Units: ft3/s or m3/s

Explanation

This FTABLE lists depth, surface area and, optionally, one or more other values (typically discharge rates) as functions of volume. HSPF interpolates between the specified values to obtain the geometric and hydraulic characteristics for intermediate values of volume.

The FTABLE must satisfy the following conditions:

1. (NCOLS*NROWS) must not exceed 100

2. NCOLS must be between 3 and 8

3. There must be at least one row in the FTABLE

4. No negative values

5. The depth and volume fields may not contain values which decrease as the row no. increases

In the example given above, we have a reach with two outflows, both of which are functions of volume. Thus, there are 5 columns in the FTABLE.

The values for this type of FTABLE can either be supplied directly by the user or be generated by a subsidiary program from more basic information (eg. by backwater analysis or Manning's equation for assumed uniform flow).

4.6 EXT SOURCES, NETWORK and EXT TARGETS Blocks

Because these blocks are very similar they are dealt with here together.

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

EXT SOURCES
<ext-svol> <exsmem> <ss><sg><-mfact--><tr> <-int-tvols--> <tgrp> <int-tmem>
              or
           <srcfmt>
...............................................................................
Above line repeats until all external sources have been specified
...............................................................................
END EXT SOURCES

NETWORK
<int-svol> <sgrp> <int-smem><-mfact--><tr> <-int-tvols--> <tgrp> <int-tmem>
...............................................................................
Above line repeats until all network entries have been made
...............................................................................
END NETWORK

EXT TARGETS
<int-svol> <sgrp> <int-smem><-mfact--><tr> <ext-tvol> <extmem> <ts> <tg> <am>
                                                         or
                                                      <tarfmt>
...............................................................................
Above line repeats until all external targets have been specified
...............................................................................
END EXT TARGETS

********************************************************************************
```

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Example
*******

EXT SOURCES
<-Volume-> <Member> SsysSgap<--Mult-->Tran <-Target vols> <-Grp> <-Member-> ***
<Name>   # <Name> # tem strg<-factor->strg <Name>   #   #        <Name> # # ***
TSS      5 INFLO    ENGL                   RCHRES   1     EXTNL  IVOL
TSS      5 ICONS    ENGL                   RCHRES   1     EXTNL  ICON
END EXT SOURCES

NETWORK
<-Volume-> <-Grp> <-Member-><--Mult-->Tran <-Target vols> <-Grp> <-Member-> ***
<Name>   #        <Name> # #<-factor->strg <Name>   #   #        <Name> # # ***
RCHRES   1 HYDR   ROVOL          0.5       RCHRES   2     EXTNL  IVOL
RCHRES   2 HYDR   ROVOL                    RCHRES   5     EXTNL  IVOL
RCHRES   4 HYDR   ROVOL                    RCHRES   5     EXTNL  IVOL
END NETWORK

EXT TARGETS
<-Volume-> <-Grp> <-Member-><--Mult-->Tran <-Volume-> <Member> Tsys Tgap Amd ***
<Name>   #        <Name> # #<-factor->strg <Name>   # <Name> #  tem strg strg***
RCHRES   5 HYDR   OVOL   2       100.      TSS     11 OUTFLO   ENGL      INST
END EXT TARGETS

********************************************************************************
```

Details

Symbol	FORTRAN Name(s)	Format	Comment
<ext-svol>	SVOL(3), SVOLNO	3A2,I4	SVOL is the external volume from which the time series come(s). Valid values are TSS (Time Series Store) and SEQ (sequential file). SVOLNO is the Dataset No., if SVOL= TSS; the Fortran unit no., if SVOL= SEQ.
<exsmem>	SMEMN(3), SMEMSB	3A2,I2	Used if SVOL= TSS. SMEMN is the name of the member, and SMEMSB is a subscript which distinguishes components belonging to the same member. Blank field(s) mean all members/subscripts are implied.
<srcfmt>	SFCLAS(3), SFNO	3A2,I2	Used if SVOL= SEQ. SFCLAS is a string indicating the class of format used in the sequential file. SFNO identifies an object-time format supplied in the FORMATS Block. Default format for the requested format class is supplied if SFNO is blank or zero.
<ss>	SSYST(2)	2A2	Unit system of data in the source. Valid values are ENGL and METR.
<sg>	SGAPST(2)	2A2	Used only if SVOL= SEQ. String indicating how missing "cards" in seq. file will be regarded. Valid values are: ZERO (assign value 0) and UNDF (assign undefined value).
<int-svol>	SVOL(3), SVOLNO	3A2,I4	SVOL is the Operation-type of the source opn. and SVOLNO is its Operation-type No. (eg. PERLND 5)
<sgrp>	SGRPN(3)	3A2	Group to which the source time series belong(s).
<int-smem>	SMEMN(3), SMEMSB(2)	3A2,2I2	Member name and subscripts identifying the source. Blank field(s) mean all members/subscripts are implied.
<mfact>	MFACTR	F10.0	The factor by which data from the source will be multiplied before being added to the target. Default (blank field)= 1.0

<tr>	TRAN(2)	2A2	String indicating which transformation function to use in transferring time series from source to target. Blank field means default value is used.
<int-tvols>	TVOL(3), TOPFST,TOPLST	3A2, 2(1X,I3)	TVOL is the Opn-type of the target. TOPFST & TOPLST specify the range of opns which are targets (eg.PERLND 1 thru 5). If TOPLST field is blank the target is a single opn. (eg. PERLND 2).
<tgrp>	TGRPN(3)	3A2	Group to which the target time series belong(s).
<int-tmem>	TMEMN(3), TMEMSB(2)	3A2,2I2	Similar to <int-smem>, but applies to target.
<ext-tvol>	TVOL(3), TVOLNO	3A2,I4	Similar to <ext-svol>, but applies to target.
<extmem>	TMEMN(3), TMEMSB	3A2,I2	Similar to <exsmem>, but applies to target.
<tarfmt>	TFCLAS(3), TFNO	3A2,I2	Similar to <srcfmt>, but applies to format of target seq file.
<ts>	TSYST(2)	2A2	Unit system in which data will be written to target.
<tg>	TGAPST(2)	2A2	Used only if TVOL= SEQ. Similar to <sg>, but applies to seq targets -- indicates whether records containing all zeros or undefined values will not be written.
<am>	AMDST(2)	2A2	Used only if TVOL= TSS. String indicating how the target dataset is to be accessed. Valid values are: ADDb, INST and REPL.

Note: All character strings must be left-justified in their fields except TSS dataset member-names (<extmem> or <exsmem>) which must be justified in the same way that they were when the dataset label was created (Section 2).

General Discussion

In these blocks the user specifies or implies those time series which are to be passed between pairs of operations in the same INGRP or between individual operations and external sources /targets (TSS Datasets or seq. files). The blocks are arranged in the form of tables, each containing one or more entries (rows). Basically, each entry contains source information, a multiplication factor and transformation info, and target information.

The entries in these blocks may be in any order.

When time series associated with datasets in the TSS are referred to, the user supplies the dataset number and, optionally, a member name and subscript. If the member information is given it must agree with data supplied when the dataset was created (Section 2.0); if the member name and/or subscript are omitted the system will assume that the entire range of time series implied by the missing data are involved. It obtains this information from the dataset label.

When data are read from or written to a sequential file the user supplies:

1. A "format class code". It fixes the nature and sequence of data in a typical record (eg. day and hr, followed by 12 hourly values).

2. The number of an object-time format, situated in the FORMATS Block. It fixes the exact format of the data in a record. A default format can be selected by supplying the number 0, or leaving the field blank.

The format classes and associated default formats presently supported in the HSPF system are documented in Sect. 4.9.

The user specifies time series which are input to, or output from, an operating module by supplying a group name (<sgrp>, <tgrp>) and, optionally, a member name plus one or two subscripts (<int-smem>, <int-tmem>). If the member info is supplied it must be compatible with data given in the Time Series Catalog for the applicable operating module and group (Section 4.7). If the member name and/or subscript(s) are not supplied the system will assume that the entire range of time series implied by the missing data are involved. It obtains this information from the Time Series Catalog.

The user may route the same source to several targets by making several separate entries in a block, each referring to the same source, or by making use of the "range" feature provided in the <int-tvols> field. (This latter feature does not apply to entries in the EXT TARGETS Block). In either case the implication is that data from the source will be used repetitively and each time will be multiplied by the specified factor and added to whatever else has already been routed to the specified target. Conversely, several sources may be routed to a single target (except in the EXT TARGETS Block). This happens when several entries specify different sources but the same target. Here, the implication is that the data obtained from the several sources must be accumulated before being used by the target.

Whenever time series are transferred from a source to a target a "transformation" takes place. The user can specify the transformation function in field <tr>; if it is blank the default function is supplied. The range of permissible functions is:

Interval relation	Kind relation	Functions: Default	Functions: Other allowable
SDELT= TDELT	* to *	SAME	none
	- to -	SAME	none
	* to -	AVER	none
SDELT> TDELT	* to *	INTP	none
	- to -	DIV	SAME
	* to -	AVER	none
SDELT< TDELT	* to *	LAST	none
	- to -	SUM	AVER, MAX, MIN
	* to -	AVER	SUM, MAX, MIN

Key:

SDELT	Time interval of source time series
TDELT	Time interval of target time series
*	Point-valued time series
-	Mean (bar)-valued time series

Note:

1. See below for an explanation of the transform keywords.

2. Keywords less than 4 characters long must be left-justified in the field.

3. For further info, see Appendix III and Time Series Catalog (Section 4.7 of this part).

Time Series Transform Functions

The time series transform functions given above are completed before the multiplication factor given in the EXTERNAL SOURCES, EXTERNAL TARGETS and NETWORK blocks are applied. These transform functions are defined as follows:

'AVER- Compute the integral of the source time series over each target time step, divide by the target time step and assign the value to the time step in the target time series. See Appendix III for definition of the integral of a time series.

DIV- Divide each mean value of the source time series by the ratio of the source time step to the target time step and assign the results to each of the target time steps contained in the source time step.

INTP- Interpolate linearly between adjacent point values in the source time series and assign the interpolated values to each time point in the target time series.

LAST- Take the value at the last time point of the source time series which belongs to the time step of the target time series and assign the value to the time step of the target time series. See Appendix III for a definition of the meaning of "belonging".

MAX- Find the maximum value of the source time series for all points belonging to the target time step (point-value time series) or find the maximum value of the source time series for all time steps contained within the target time step (mean-value time series). Assign the maximum value to the time step of the target time series. The definition of "belonging" (given in Appendix III) was motivated by the desire to make MAX and MIN unique for point-value time series.

MIN- Find the minimum value of the source time series for all points belonging to the target time step (point-value time series) or find the minimum value of the source time series for all time steps contained within the target time series (mean-value time series). Assign the minimum value to the time step of the target time series.

SAME- Take the value at each time step or time point of the source time series and assign th value to the corresponding time point (point- value time series), the corresponding time step (mean-value time series), or all the contained time steps (mean-value time series with time step less than the source time step) of the target time series.

SUM- For point-value source time series: Compute the sum of the values for all points in the source time series belonging to the target series time step plus the value of the source time series at the initial point of the target time step and assign the sum to the target time step. For mean-value source time series: Compute the sum of the values for all time steps in the source time series contained within the target series time step and assign the sum to the target time step.

EXT SOURCES Block

In this block the user specifies or implies those time series which are to be supplied to operations in a RUN from sources external to it (from TSS Datasets or sequential files). This block must always be present in a RUN Data Set, because every operation must have one or more time series input to it.

If an entry specifies the source volume as SEQ, the user is referring to a single time series coming from a sequential file. He therefore must supply the Fortran unit no. and format information for the file.

If an entry specifies the source volume as TSS the user may be referring to a single time series or a group, as discussed earlier. The same applies to the target information.

NETWORK Block

In this block the user specifies or implies those time series which will be passed between operations via the internal scratch pad (INPAD). If there are no such linkages the block is omitted.

The example above shows how this block is used to specify the connectivity of a set of reaches of stream channel (RCHRES 1 flows to RCHRES 2, RCHRES 2 and 4 flow to RCHRES 5). It can also be used to specify the flow of time series data from utility operations to simulation operations and vice versa. The network can be extremely complex, or non-existent (eg. if the RUN involves only one operation).

Because the time series are transferred via the INPAD each source and target pair must be in the same INGRP.

EXT TARGETS Block

In this block the user specifies or implies those time series which will be output from the operations in a RUN, to datasets in the TSS or to sequential files. If there are no such transfers the block is omitted.

This block is similar to the EXT SOURCES Block but serves the opposite purpose. Thus, the entries have similar formats (but are reversed). In addition, each entry in the EXT TARGETS Block has the <am> field, which indicates how the target dataset will be accessed, if it is in the TSS. The valid values and the meaning of each are:

ADD This option preserves pre-existing data (in the TSS dataset) which precedes the starting time of the RUN. Pre-existing data subsequent to that time, including any which goes beyond the ending time of the RUN, is destroyed. The year order option (YEAROR), specified when the dataset label was created or updated, must be YES.

INST This option is used to write data to the dataset for calendar years for which no data pre-exists. No pre-existing data are changed or destroyed, and YEAROR need not be YES.

REPL This option preserves pre-existing data both before and after the time span of the RUN. Data in the dataset must be in uncompressed form (COMPRESSION= UNCOMP). Because this option is designed for replacement of data, some data must pre-exist for every calendar year of the replacement period (RUN span).

Option ADD or INST must be used when time series data are first placed in a data set. ADD will result in data for every calendar year being physically positioned (in the TSS dataset) in chronological order; INST will not. Note, however, that within each calendar year data are always stored in chronological order. REPL must be used if the data are to be selectively changed, without affecting data outside the period of change.

WARNINGS!

1. Some complications arise if you are dealing with a multi-component TSS dataset. For example, if you have previously read data into one or more of the components and you now wish to read data into other components, REPL must be used. If ADD were used, the existing data would be wiped out, even though they are not stored in components used in the present run. ADD overwrites data in all components of a dataset. Similarly, INST could not be used. Even if the components being written to in the present run are empty, the presence of data in the other components would preclude the use of INST. This option requires that all the components be empty for all calendar years covered by the run.

2. In this block it is not permissible to route several sources to the same target. If you want to combine several time series and write the result to an external target, first use a utility operation (COPY) to combine the data and then use this block to route the result to the external target.

3. The same applies if you need to store several time series as components of the same TSS dataset. Collect them first, using COPY Module, and output them all to the TSS with a single entry in the EXT TARGETS Block.

4. It is dangerous to refer to the same TSS dataset in both the EXT SOURCES and EXT TARGETS Blocks. That is, you must not try to both read from and write to the same dataset in one run.

5. If the above warnings are not heeded, you may cause irreparable damage to your TSS.

4.7 Time Series Catalog

This section documents all the time series which are required by, and which can be output by, all the operating modules in the HSPF system.

The time series are arranged in groups. Thus, to specify an operation associated time series in the EXT SOURCES, NETWORK or EXT TARGETS Blocks, the user supplies a group name followed, optionally, by a member name and subscripts.

The time series documented in this section can be separated into three categories:

1. Input only. Some time series can only be input to their operating module (eg. member PREC of group EXTNL in module PERLND).

2. Input or output. Some time series can either be input to their operating module or output from it, depending on the options in effect. For example, if snow accum and melt on a Pervious Land-segment (PLS) is being simulated in a given RUN, time series WYIELD in group SNOW can be output to the Time Series Store (TSS). Then, if section SNOW is inactive but section PWATER is active in a subsequent RUN, the same time series WYIELD may be specified as an input to the PERLND module. This feature makes it possible to calibrate an application module in an incremental manner. First, the outputs from section 1 are calibrated to the field data; then the outputs from section 2 are calibrated using outputs from section 1 as inputs, etc. Sections calibrated in earlier runs need not be re-run if the needed outputs from them have been stored.

3. Output only. Some time series can be computed by and output from their operating module, but never serve as inputs to it (eg. member ALBEDO of group SNOW in module PERLND).

To run an operating module, the user must ensure that all the input time series which it requires are made available to it. He does this by making appropriate entries in the EXT SOURCES or NETWORK blocks. To ascertain which time series are required, he should consult the Time Series Catalog for the appropriate module. For example, suppose sediment production and washoff/ scour from a PLS are being simulated using snowmelt and water budget results from a previous RUN. That is, section SEDMNT is active but sections ATEMP, SNOW and PWATER are not. Then, Table 4.7(1).5 shows:

1. member PREC of group EXTNL is a required input time series (member SLSED is optional)

2. members RAINF and SNOCOV of group SNOW are reqd inputs, because section SNOW is inactive

3. members SURO and SURS of group PWATER are reqd inputs, because section PWATER is inactive (SUROB and SURSB may also be reqd)

The user can obtain further details on the above time series by consulting the table for the appropriate group (eg. Table 4.7(1).1 for group EXTNL).

Table 4.7(1).5 shows which time series are computed in the SEDMNT section of the PERLND module and may therefore be output (members DETS through SOSDB).

Thus, in the EXT SOURCES and/or NETWORK blocks, entries must appear which specify members PREC, RAINF, etc (groups EXTNL, SNOW, PWATER) as targets to which source time series are routed. Also, in the NETWORK and/or EXT TARGETS blocks, entries may appear which specify one or more of members DETS through SOSDB (of group SEDMNT) as source time series, which are routed to other operations or to the TSS or a sequential file.

The tables which follow are otherwise self explanatory, except for the abbreviation "ivld" which appears frequently in the "Units" fields. It means "interval of the data" (to distinguish it from the internal, or simulation interval). Thus, if a TSS dataset containing 1-hour precip data is input to an operation with a DELT of two hours, ivld is 1 hour.

4.7(1) Catalog for PERLND module

The time series groups associated with this application module are shown in Figure 4.7(1)-1.

The members contained within each group are documented in the tables which follow.

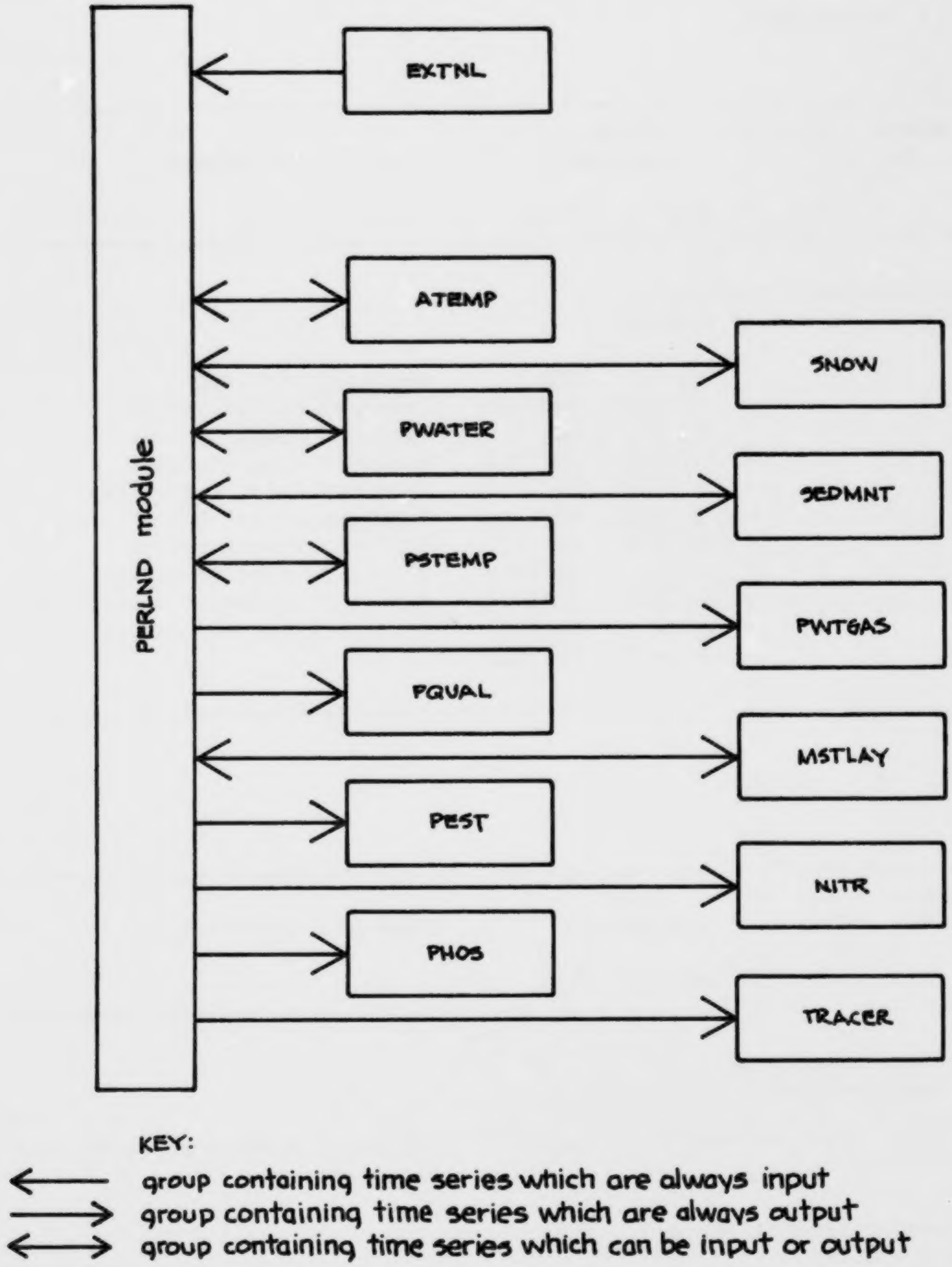

PERLND module
EXTNL
ATEMP
SNOW
PWATER
SEDMNT
PSTEMP
PWTGAS
PQUAL
MSTLAY
PEST
NITR
PHOS
TRACER
KEY:
group containing time series which are always input
group containing time series which are always output
group containing time series which can be input or output

Figure 4.7(1)-1 Groups of time series associated with the PERLND module

4.7(1).1 Group EXTNL

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series always external (input only) to module PERLND:						
GATMP	1	1	-	Deg F	Deg C	Measured air temp
PREC	1	1	-	in/ivld	mm/ivld	Measured precipitation
DTMPG	1	1	-	Deg F	Deg C	Measured dewpoint temp
WINMOV	1	1	-	mi/ivld	km/ivld	Measured wind movement
SOLRAD	1	1	-	Ly/ivld	Ly/ivld	Measured solar radiation
PETINP	1	1	-	in/ivld	mm/ivld	Input potential E-T
SURLI	1	1	-	in/ivld	mm/ivld	Surface lateral inflow
IFWLI	1	1	-	in/ivld	mm/ivld	Interflow lateral inflow
AGWLI	1	1	-	in/ivld	mm/ivld	Active groundwater lateral inflow
SLSED	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Lateral input of sediment

4.7(1).2 Group ATEMP

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section ATEMP:						
AIRTMP	1	1	-	Deg F	Deg C	Estimated surface air temp over the Land-segment
Input time series reqd to compute the above:						
Group EXTNL						always required
GATMP						
PREC						

4.7(1).3 Group SNOW

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section SNOW:						
PACK	1	1	*	in	mm	Total contents of pack(water equiv)
PACKF	1	1	*	in	mm	Frozen contents of pack, ie. snow+ ice (water equiv)
PACKW	1	1	*	in	mm	Liquid water in pack
PACKI	1	1	*	in	mm	Ice in pack (water equiv)
PDEPTH	1	1	*	in	mm	Pack depth
RDENPF	1	1	*	none	none	Relative density of frozen contents of pack (PACKF/PDEPTH)
SNOCOV	1	1	*	none	none	Fraction of Land-segment covered by pack
ALBEDO	1	1	*	none	none	Albedo of the pack
PAKTMP	1	1	*	Deg F	Deg C	Mean temp of the pack
SNOWF	1	1	-	in/ivld	mm/ivld	Snowfall, water equivalent
SNOWE	1	1	-	in/ivld	mm/ivld	Evap from PACKF (sublimation), water equivalent
WYIELD	1	1	-	in/ivld	mm/ivld	Water yielded by the pack (released to the land-surface)
MELT	1	1	-	in/ivld	mm/ivld	Quantity of melt from PACKF (water equivalent)
RAINF	1	1	-	in/ivld	mm/ivld	Rainfall

Input time series reqd to compute the above:

Group EXTNL	always required
PREC	
DTMPG	
WINMOV	
SOLRAD	
Group ATEMP	only reqd if section ATEMP inactive
AIRTMP	

4.7(1).4 Group PWATER

<---- Member ---->			Kind	Units (external)		Description/comment
Name	Max subscr values 1	2		Engl	Metr	
Time series computed by module section PWATER:						
Land-segment-wide values:						
PERS	1	1	*	in	mm	Total water stored in the PLS
CEPS	1	1	*	in	mm	Interception storage
SURS	1	1	*	in	mm	Surface (overland flow) storage
UZS	1	1	*	in	mm	Upper zone storage
IFWS	1	1	*	in	mm	Interflow storage
LZS	1	1	*	in	mm	Lower zone storage
AGWS	1	1	*	in	mm	Active groundwater storage
RPARM	1	1	-	in/ivld	mm/ivld	Current value of max lower zone E-T opportunity
SURO	1	1	-	in/ivld	mm/ivld	Surface outflow
IFWO	1	1	-	in/ivld	mm/ivld	Interflow outflow
AGWO	1	1	-	in/ivld	mm/ivld	Active groundwater outflow
PERO	1	1	-	in/ivld	mm/ivld	Total outflow from PLS
IGWI	1	1	-	in/ivld	mm/ivld	Inflow to inactive (deep) groundwater
PET	1	1	-	in/ivld	mm/ivld	Potential E-T, adjusted for snow cover and air temp
CEPE	1	1	-	in/ivld	mm/ivld	Evap from interception storage
UZET	1	1	-	in/ivld	mm/ivld	E-T from upper zone
LZET	1	1	-	in/ivld	mm/ivld	E-T from lower zone
AGWET	1	1	-	in/ivld	mm/ivld	E-T from active groundwater storage
BASET	1	1	-	in/ivld	mm/ivld	E-T taken from active groundwater outflow (baseflow)
TAET	1	1	-	in/ivld	mm/ivld	Total simulated E-T
IFWI	1	1	-	in/ivld	mm/ivld	Interflow inflow (excluding any lateral inflow)
UZI	1	1	-	in/ivld	mm/ivld	Upper zone inflow
INFIL	1	1	-	in/ivld	mm/ivld	Infiltration to the soil
PERC	1	1	-	in/ivld	mm/ivld	Percolation from upper to lower zone
LZI	1	1	-	in/ivld	mm/ivld	Lower zone inflow
AGWI	1	1	-	in/ivld	mm/ivld	Active groundwater inflow (excl any lateral inflow)
SURI	1	1	-	in/ivld	mm/ivld	Surface inflow (including any lateral inflow)

Block-specific values:

SURSB	NBLKS	1	*	in	mm	Surface storage
UZSB	NBLKS	1	*	in	mm	Upper zone storage
IFWSB	NBLKS	1	*	in	mm	Interflow storage
SUROB	NBLKS	1	-	in/ivld	mm/ivld	Surface outflow
IFWOB	NBLKS	1	-	in/ivld	mm/ivld	Interflow outflow
UZETB	NBLKS	1	-	in/ivld	mm/ivld	Upper zone E-T
IFWIB	NBLKS	1	-	in/ivld	mm/ivld	Interflow inflow (excl any lateral inflow)
UZIB	NBLKS	1	-	in/ivld	mm/ivld	Upper zone inflow
INFILB	NBLKS	1	-	in/ivld	mm/ivld	Infiltration
PERCB	NBLKS	1	-	in/ivld	mm/ivld	Percolation from upper to lower zone

Input time series reqd to
compute the above:

Group EXTNL

SURLI	optional
IFWLI	optional
AGWLI	optional
PETINP	
PREC	reqd if snow not considered (CSNOFG= 0)

Group ATEMP

AIRTMP	only reqd if section ATEMP inactive and CSNOFG= 1

Group SNOW	only reqd if section SNOW inactive and snow is considered (CSNOFG= 1)
RAINF	
SNOCOV	
WYIELD	
PACKI	only reqd if ICEFG= 1

4.7(1).5 Group SEDMNT

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section SEDMNT:						
Land-segment-wide values:						
DETS	1	1	*	tons/ac	tonnes/ha	Storage of detached sediment
WSSD	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Washoff of detached sediment
SCRSD	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Scour of matrix (attached) soil
SOSED	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Total removal of soil and sediment
DET	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Quantity of sediment detached from soil matrix by rainfall impact
Block-specific values:						
DETSB	NBLKS	1	*	tons/ac	tonnes/ha	Storage of detached sediment
WSSDB	NBLKS	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Washoff of detached sediment
SCRSDB	NBLKS	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Scour of matrix (attached) soil
SOSDB	NBLKS	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Total removal of soil and sediment from block
Input time series reqd to compute the above:						
Group EXTNL						always required
PREC						
SLSED						optional
Group SNOW						only reqd if section SNOW inactive and snow considered (CSNOFG= 1)
RAINF						
SNOCOV						
Group PWATER						only reqd if sect PWATER inactive
SURO						
SURS						
SUROB						only reqd if NBLKS> 1
SURSB						only reqd if NBLKS> 1

4.7(1).6 Group PSTEMP

Name (Member)	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section PSTEMP:						
AIRTC	1	1	*	Deg F	Deg C	Air temp on the PLS
SLTMP	1	1	*	Deg F	Deg C	Surface layer soil temp
ULTMP	1	1	*	Deg F	Deg C	Upper layer soil temp
LGTMP	1	1	*	Deg F	Deg C	Lower and groundwater layer soil temp
Input time series required to compute the above:						
Group ATEMP						only reqd if section ATEMP inactive
AIRTMP						

4.7(1).7 Group PWTGAS

Name (Member)	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section PWTGAS:						
SOTMP	1	1	*	Deg F	Deg C	Temp of surface outflow
IOTMP	1	1	*	Deg F	Deg C	Temp of interflow outflow
AOTMP	1	1	*	Deg F	Deg C	Temp of active groundwater outflow
SODOX	1	1	*	mg/l	mg/l	DO conc in surface outflow
SOCO2	1	1	*	mg/l	mg/l	CO2 conc in surface outflow
IODOX	1	1	*	mg/l	mg/l	DO conc in interflow outflow
IOCO2	1	1	*	mg/l	mg/l	CO2 conc in interflow outflow
AODOX	1	1	*	mg/l	mg/l	DO conc in active groundw outflow

AOCO2	1	1	*	mg/l	mg/l	CO2 conc in active groundw outflow
SOHT	1	1	-	BTU/ac.ivld	kcal/ha.ivld	Heat energy in surface outflow (relative to freezing point)
IOHT	1	1	-	BTU/ac.ivld	kcal/ha.ivld	Heat energy in interflow outflow
AOHT	1	1	-	BTU/ac.ivld	kcal/ha.ivld	Heat energy in active groundwater outflow
POHT	1	1	-	BTU/ac.ivld	kcal/ha.ivld	Heat energy in total outflow from PLS
SODOXM	1	1	-	lb/ac.ivld	kg/ha.ivld	Flux of DO in surface outflow
SOCO2M	1	1	-	lb/ac.ivld	kg/ha.ivld	Flux of CO2 in surface outflow
IODOXM	1	1	-	lb/ac.ivld	kg/ha.ivld	Flux of DO in interflow outflow
IOCO2M	1	1	-	lb/ac.ivld	kg/ha.ivld	Flux of CO2 in interflow outflow
AODOXM	1	1	-	lb/ac.ivld	kg/ha.ivld	Flux of DO in active groundwater outflow
AOCO2M	1	1	-	lb/ac.ivld	kg/ha.ivld	Flux of CO2 in active groundwater outflow
PODOXM	1	1	-	lb/ac.ivld	kg/ha.ivld	DO in total outflow from PLS
POCO2M	1	1	-	lb/ac.ivld	kg/ha.ivld	CO2 in total outflow from PLS

Input time series reqd to
compute the above:

Group SNOW	only reqd if section SNOW inactive and snow considered (CSNOFG= 1)
WYIELD	
Group PWATER	only reqd if sect PWATER inactive
SURO IFWO AGWO	
Group PSTEMP	only reqd if sect PSTEMP inactive
SLTMP ULTMP LGTMP	

4.7(1).8 Group PQUAL

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section PQUAL:						
SQO	NQOF	1	*	qty/ac	qty/ha	Storage of QUALOF on the surface
WASHQS	NQSD	1	-	qty/ ac.ivld	qty/ ha.ivld	Removal of QUALSD by assoc with detached sed washoff
SCRQS	NQSD	1	-	qty/ ac.ivld	qty/ ha.ivld	Removal of QUALSD by assoc with scour of matrix soil
SOQS	NQSD	1	-	qty/ ac.ivld	qty/ ha.ivld	Total flux of QUALSD from surface
SOQO	NQOF	1	-	qty/ ac.ivld	qty/ ha.ivld	Washoff of QUALOF from surface
SOQUAL	NQ	1	-	qty/ ac.ivld	qty/ ha.ivld	Total outflow of QUAL from the surface
IOQUAL	NQIF	1	-	qty/ ac.ivld	qty/ ha.ivld	Outflow of QUAL in interflow (QUALIF)
AOQUAL	NQGW	1	-	qty/ ac.ivld	qty/ ha.ivld	Outflow of QUAL in active groundwater outflow (QUALGW)
POQUAL	NQ	1	-	qty/ ac.ivld	qty/ ha.ivld	Total flux of QUAL from the PLS
SOQOC	NQOF	1	-	qty/ft3	qty/l	Conc of QUALOF in surface outflow
SOQC	NQ	1	-	qty/ft3	qty/l	Conc of QUAL (QUALSD+QUALOF) in surface outflow
POQC	NQ	1	-	qty/ft3	qty/l	Conc of QUAL (total) in total outflow from PLS
Input time series reqd to compute the above:						
Group PWATER						only reqd if sect PWATER inactive
SURO						only reqd if one or more QUALs are QUALOFs, or if SOQC is reqd for one or more QUALs
IFWO						only reqd if one or more QUALs are QUALIFs
AGWO						only reqd if one or more QUALs are QUALGWs
PERO						only reqd if POQC is reqd for one or more QUALs
Group SEDMNT WSSD SCRSD						only reqd if sect SEDMNT inactive and one or more QUALs are QUALSDs

4.7(1).9 Group MSTLAY

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series computed by module section MSTLAY:						
MST	5	1	*	lb/ac	kg/ha	Water in surface, upper princ, upper auxil, lower and groundw storages (segment-wide values)
FRAC	8	1	*	/ivl	/ivl	Fractional fluxes thru soil: FSO,FSP,FII,FUP,FIO,FLP,FLDP,FAO (segment-wide values)
MSTB	3	NBLKS	*	lb/ac	kg/ha	Water in surface, upper princ, and upper auxil storages (block-specific values)
FRACB	5	NBLKS	*	lb/ac	kg/ha	Fractional fluxes thru topsoil layers: FSOB,FSPB,FIIB,FUPB,FIOB (block-specific values)
Input time series reqd to compute the above:						
Group PWATER:						only reqd if sect PWATER inactive
SURI,LZS,IGWI,AGWI,AGWS,AGWO						
SURS,SURO,INFIL,IFWI,UZI,UZS,PERC, IFWS,IFWO						only reqd if NBLKS=1
SURSB,SUROB,INFILB,IFWIB,UZIB,UZSB, PERCB,IFWSB,IFWOB						only reqd if NBLKS>1

4.7(1).10 Group PEST

<---- Member ---->			K	Units (external)		Description/comment
Name	Max subscr values 1	2	ind	Engl	Metr	

Time series computed by module section PEST:

Name	1	2	Kind	Engl	Metr	Description/comment
SPS	3	NPST	*	lb/ac	kg/ha	Amount of pesticide in surface storage
UPS	3	NPST	*	lb/ac	kg/ha	Amount of pesticide in upper principal storage
IPS	NPST	1	*	lb/ac	kg/ha	Amount of pesticide in upper auxil (interflow) storage
LPS	3	NPST	*	lb/ac	kg/ha	Amount of pesticide in lower layer storage
APS	3	NPST	*	lb/ac	kg/ha	Amount of pesticide in active groundwater layer storage
TPS	3	NPST	*	lb/ac	kg/ha	Total amount of pesticide in the soil

Note: SPS,UPS,LPS,APS and TPS give the storage of each pesticide by species. The first subscript indicates the species; crystalline, adsorbed or solution, The second indicates the pesticide.
For example, UPS(2,3) is the quantity of adsorbed pesticide in the upper layer principal storage, for the 3rd pesticide.
The first subscript for IPS has a max value of one because only solution pesticide is modelled in the upper layer auxil (interflow) layer.

Name	1	2	Kind	Engl	Metr	Description/comment
TOTPST	NPST	1	*	lb/ac	kg/ha	Total amount of pesticide in the soil (sum of all species).
SDPS	2	NPST	-	lb/ac.ivld	kg/ha.ivld	Outflow of sediment-associated pesticide (SDPSY and SDPSA for each pesticide)
TSPSS	5	NPST	-	"	"	Fluxes of solution pesticide for the topsoil layers: SOPSS,SPPSS, UPPSS,IIPSS,IOPSS
SSPSS	3	NPST	-	"	"	Fluxes of solution pesticide for the subsoil layers: LPPSS,LDPPSS, AOPSS
SDEGPS	NPST	1	-	"	"	Amount of degradation in surf. layer
UDEGPS	NPST	1	-	"	"	Amount of degradation in upper layer
LDEGPS	NPST	1	-	"	"	Amount of degradation in lower layer
ADEGPS	NPST	1	-	"	"	Amount of degr. in groundw. layer
TDEGPS	NPST	1	-	"	"	Total amount of degradation in soil
SOSDPS	NPST	1	-	"	"	Total outflow of sediment-associated pesticide (SDPSY + SDPSA)

POPST	NPST	1	-	"	"	Total outflow of solution and sed-associated pesticide from the PLS

Note: The subscript with max value NPST selects the particular pesticide. For example, POPST(2,1) is the total outflow from the PLS of the second pesticide.

Input time series reqd to
compute the above:

Group SEDMNT	only reqd if section SEDMNT inactive
SOSED	required if NBLKS=1
SOSEDB	required if NBLKS>1

Group PSTEMP	only reqd if section PSTEMP inactive
SLTMP	
ULTMP	
LGTMP	

Group MSTLAY	only reqd if section MSTLAY inactive
If NBLKS=1:	
MST	
FRAC	
If NBLKS>1:	
MST(4&5)	
FRAC(6 thru 8)	
MSTB	
FRACB	

4.7(1).11 Group NITR

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment

Time series computed by
module section NITR:

Name	1	2	Kind	Engl	Metr	Description/comment
SN	5	1	*	lb/ac	kg/ha	N in surface layer storage
UN	5	1	*	"	"	N in upper layer princ storage
LN	5	1	*	"	"	N in lower layer storage
AN	5	1	*	"	"	N in groundwater layer storage
TN	5	1	*	"	"	Total N in soil, by species

In the above, the first subscript selects the species of N:
1 means organic N, 2 means adsorbed ammonium, 3 means solution ammonium,
4 means nitrate, 5 means plant N derived from this layer

Name	1	2	Kind	Engl	Metr	Description/comment
IN	2	1	*	lb/ac	kg/ha	N in upper layer auxil (interflow) storage

In the above, the first subscript selects the species of N:
1 means solution ammonium, 2 means nitrate
(only soluble species are modelled in this storage)

Name	1	2	Kind	Engl	Metr	Description/comment
TOTNIT	1	1	*	lb/ac	kg/ha	Total N stored in the PLS (all species)
SEDN	2	1	-	lb/ac.ivld	kg/ha.ivld	Outflows of sediment-associated N

In the above, the first subscript selects the flux:
1 means organic N removal, 2 means adsorbed ammonium removal

Name	1	2	Kind	Engl	Metr	Description/comment
SOSEDN	1	1	-	lb/ac.ivld	kg/ha.ivld	Total outflow of sediment-associated N (orgN + ads ammon)
TSAMS	5	1	-	lb/ac.ivld	kg/ha.ivld	Fluxes of soln ammon in the topsoil
TSNO3	5	1	-	"	"	Fluxes of nitrate in the topsoil

In the above, the first subscript selects the flux:
1 means outflow with surface water outflow
2 means percolation from surface to upper layer principal storage
3 means percolation from upper layer principal storage to lower layer storage
4 means flow from upper layer principal to upper layer auxil (interflow) storage
5 means outflow from PLS with water from upper layer auxil (interflow) storage

SSAMS	3	1	-	lb/ac.ivld	kg/ha.ivld	Fluxes of soln ammon in the subsoil
SSNO3	3	1	-	"	"	Fluxes of nitrate in the subsoil

In the above, the first subscript selects the flux:
1 means percolation from the lower layer to the active groundwater storage
2 means deep percolation, from the lower layer to inactive groundwater
3 means outflow from the PLS with water from the active groundw storage

PONO3	1	1	-	lb/ac.ivld	kg/ha.ivld	Total outflow of NO3 from the PLS
PONH4	1	1	-	"	"	Total outflow of NH4 from the PLS
PONITR	1	1	-	"	"	Total outflow of N (NO3+NH4+ORGN) from the PLS.
TDENIF	1	1	-	"	"	Total denitrification in the PLS

Input time series reqd to
compute the above:

Same as for section PEST. An input time series need only be supplied if section PEST and the section which computes it (SEDMNT, PSTEMP or MSTLAY) are inactive.

4.7(1).12 Group PHOS

<---- Member ----> Pame	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment

Time series computed by
module section PHOS:

SP	4	1	*	lb/ac	kg/ha	P in surface layer storage
UP	4	1	*	"	"	P in upper layer princ storage
LP	4	1	*	"	"	P in lower layer storage
AP	4	1	*	"	"	P in groundwater layer storage
TP	4	1	*	"	"	Total P in soil, by species

In the above, the first subscript selects the species of P:
1 means organic P, 2 means adsorbed phosphate, 3 means solution phosphate,
4 means plant P derived from this layer

IP	1	1	*	lb/ac	kg/ha	P in upper layer auxil (interflow) storage (solution phosphate)

(only soluble species are modelled in this storage)

TOTPHO	1	1	*	lb/ac	kg/ha	Total P stored in the PLS (all species)
SEDP	2	1	-	lb/ac.ivld	kg/ha.ivld	Outflows of sediment-associated P

In the above, the first subscript selects the flux:
1 means organic P removal, 2 means adsorbed phosphate removal

SOSEDP	1	1	-	lb/ac.ivld	kg/ha.ivld	Total outflow of sediment-associated P (orgP + ads phosphate)
TSP4S	5	1	-	lb/ac.ivld	kg/ha.ivld	Fluxes of soln phosphate in the topsoil.

In the above, the first subscript selects the flux:
1 means outflow with surface water outflow
2 means percolation from surface to upper layer principal storage
3 means percolation from upper layer principal storage to lower layer storage
4 means flow from upper layer principal to upper layer auxil (interflow) storage
5 means outflow from PLS with water from upper layer auxil (interflow) storage

SSP4S	3	1	-	lb/ac.ivld	kg/ha.ivld	Fluxes of soln phosphate in the subsoil.

In the above, the first subscript selects the flux:
1 means percolation from the lower layer to the active groundwater storage
2 means deep percolation, from the lower layer to inactive groundwater
3 means outflow from the PLS with water from the active groundw storage

POPHOS	1	1	-	"	"	Total outflow of P from the PLS.

Input time series reqd to
compute the above:

Same as for section PEST. An input time series need only be supplied if sections PEST and NITR and the module section which computes it (SEDMNT, PSTEMP or MSTLAY) are inactive.

4.7(1).13 Group TRACER

Time series computed by
module section TRACER:

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
STRSU	1	1	*	lb/ac	kg/ha	Tracer in surface layer storage
UTRSU	1	1	*	"	"	Tracer in upper layer princ storage
ITRSU	1	1	*	"	"	Tracer in upper layer auxil storage
LTRSU	1	1	*	"	"	Tracer in lower layer storage
ATRSU	1	1	*	"	"	Tracer in groundw layer storage
TRSU	1	1	*	"	"	Total tracer stored in the PLS
TSTRS	5	1	-	lb/ac.ivld	kg/ha.ivld	Fluxes of tracer in topsoil

In the above, the first subscript indicates the flux:
1 means outflow with surface water outflow
2 means percolation from surface to upper layer principal storage
3 means percolation from upper layer principal to lower layer storage
4 means flow from upper principal to upper auxil (interflow) storage
5 means outflow from the PLS from upper layer transitory (interflow) storage

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
SSTRS	3	1	-	lb/ac.ivld	kg/ha.ivld	Fluxes of tracer in subsoil

In the above, the first subscript indicates the flux:
1 means percolation from lower layer to active groundwater storage
2 means deep percolation, from lower layer to inactive groundwater
3 means outflow from the PLS from the active groundwater storage

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
POTRS	1	1	-	lb/ac.ivld	kg/ha.ivld	Total outflow of tracer from the PLS

Input time series reqd to
compute the above:

Name	Description/comment
Group MSTLAY If NBLKS=1: MST FRAC If NBLKS>1: MST(4&5) FRAC(6 thru 8) MSTB FRACB	only reqd if MSTLAY, PEST, NITR and PHOS are all inactive; else these time series will already have been supplied

4.7(2) Catalog for IMPLND module

The time series groups associated with this application module are shown in Figure 4.7(2)-1.

The members contained within each group are documented in the tables which follow.

4.7(2).1 Group EXTNL

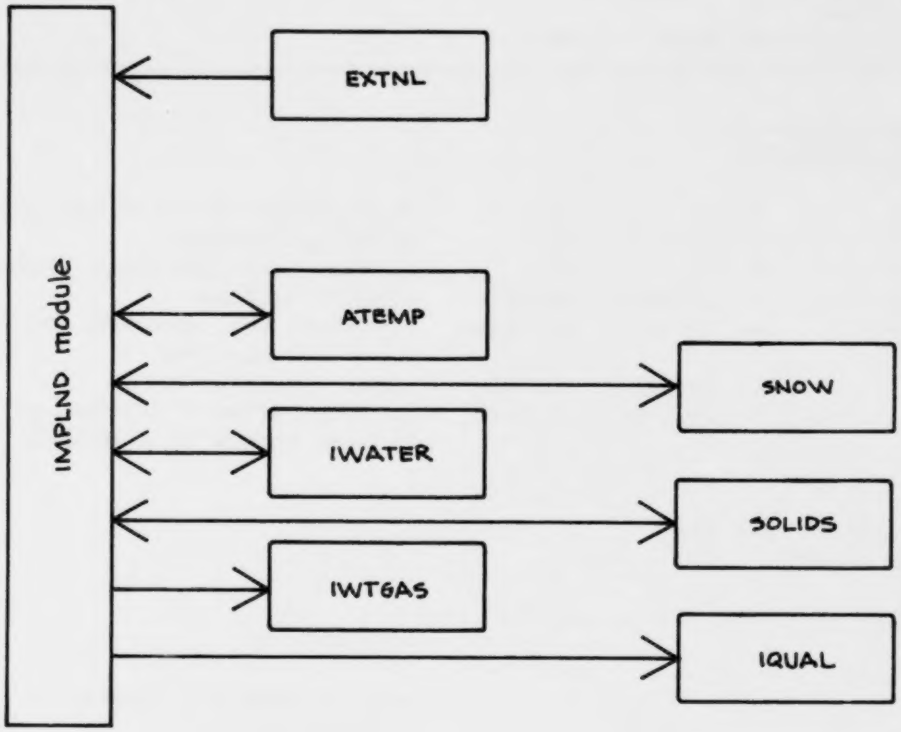
<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series always external (input only) to module IMPLND:						
GATMP	1	1	-	Deg F	Deg C	Measured air temp
PREC	1	1	-	in/ivld	mm/ivld	Measured precipitation
DTMPG	1	1	-	Deg F	Deg C	Measured dewpoint temp
WINMOV	1	1	-	mi/ivld	km/ivld	Measured wind movement
SOLRAD	1	1	-	Ly/ivld	Ly/ivld	Measured solar radiation
PETINP	1	1	-	in/ivld	mm/ivld	Input potential E-T
SURLI	1	1	-	in/ivld	mm/ivld	Surface lateral inflow
SLSLD	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Lateral input of solids

4.7(2).2 Group ATEMP

Identical to group ATEMP in module PERLND. See Section 4.7(1).2 for documentation.

4.7(2).3 Group SNOW

Identical to group SNOW in module PERLND. See Section 4.7(1).3 for documentation.



KEY:

← group containing time series which are always input

→ group containing time series which are always output

↔ group containing time series which can be input or output

Figure 4.7(2)-1 Groups of time series associated with the IMPLND module

4.7(2).4 Group IWATER

<---- Member ---->			K	Units		
	Max subscr		i	(external)		Description/comment
Name	values		n			
	1	2	d	Engl	Metr	
Time series computed by module section IWATER:						
IMPS	1	1	*	in	mm	Total water stored in the ILS
RETS	1	1	*	in	mm	Retention storage
SURS	1	1	*	in	mm	Surface (overland flow) storage
SURO	1	1	-	in/ivld	mm/ivld	Surface outflow
PET	1	1	-	in/ivld	mm/ivld	Potential E-T, adjusted for snow cover and air temp
IMPEV	1	1	-	in/ivld	mm/ivld	Total simulated E-T
SURI	1	1	-	in/ivld	mm/ivld	Surface inflow (including any lateral inflow if RTLIFG=1)
Input time series reqd to compute the above:						
Group EXTNL						
SURLI						optional
PETINP						
PREC						reqd if snow not considered (CSNOFG= 0)
Group ATEMP						
AIRTMP						only reqd if section ATEMP inactive and CSNOFG= 1
Group SNOW						only reqd if section SNOW inactive and snow is considered (CSNOFG= 1)
RAINF						
SNOCOV						
WYIELD						

4.7(2).5 Group SOLIDS

<---- Member ---->			Kind	Units (external)		Description/comment
Name	Max subscr values 1	2		Engl	Metr	
Time series computed by module section SOLIDS:						
SLDS	1	1	*	tons/ac	tonnes/ha	Storage of solids on surface
SOSLD	1	1	-	tons/ ac.ivld	tonnes/ ha.ivld	Washoff of solids from surface
Input time series reqd to compute the above:						
Group EXTNL						always required
PREC						
SLSLD						optional
Group IWATER						only reqd if sect IWATER inactive
SURO						
SURS						

4.7(2).6 Group IWTGAS

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series computed by module section IWTGAS:						
SOTMP	1	1	*	Deg F	Deg C	Temp of surface outflow
SODOX	1	1	*	mg/l	mg/l	DO conc in surface outflow
SOCO2	1	1	*	mg/l	mg/l	CO2 conc in surface outflow
SOHT	1	1	-	BTU/ ac.ivld	kcal/ ha.ivld	Heat energy in surface outflow (relative to freezing point)
SODOXM	1	1	-	lb/ ac.ivld	kg/ ha.ivld	Flux of DO in surface outflow
SOCO2M	1	1	-	lb/ ac.ivld	kg/ ha.ivld	Flux of CO2 in surface outflow
Input time series reqd to compute the above:						
Group ATEMP						only reqd if section ATEMP inactive
AIRTMP						
Group SNOW						only reqd if section SNOW inactive and snow considered (CSNOFG= 1)
WYIELD						
Group IWATER						only reqd if sect IWATER inactive
SURO						

4.7(2).7 Group IQUAL

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series computed by module section IQUAL:						
SQO	NQOF	1	*	qty/ac	qty/ha	Storage of QUALOF on the surface
SOQS	NQSD	1	-	qty/ ac.ivld	qty/ ha.ivld	Total flux of QUALSD from surface
SOQO	NQOF	1	-	qty/ ac.ivld	qty/ ha.ivld	Washoff of QUALOF from surface
SOQUAL	NQ	1	-	qty/ ac.ivld	qty/ ha.ivld	Total outflow of QUAL from the surface
SOQOC	NQOF	1	-	qty/ft3	qty/l	Conc of QUALOF in surface outflow
SOQC	NQ	1	-	qty/ft3	qty/l	Conc of QUAL (QUALSD+QUALOF) in surface outflow
Input time series reqd to compute the above:						
Group IWATER						only reqd if sect IWATER inactive
SURO						only reqd if one or more QUALs are QUALOFs, or if SOQC is reqd for one or more QUALs
Group SOLIDS						only reqd if sect SOLIDS inactive
SOSLD						and one or more QUALs are QUALSDs

4.7(3) Catalog for RCHRES module

The time series groups associated with this application module are shown in Figure 4.7(3)-1.

The members contained within each group are documented in the tables which follow.

4.7(3).1 Group EXTNL

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series external to module RCHRES (input only):						
PREC	1	1	-	in/ivld	mm/ivld	Precip on surface of the RCHRES
POTEV	1	1	-	in/ivld	mm/ivld	Potential evap from the surface
COLIND	NEXITS	1	-	none	none	Time series indicating which (pair of) columns in RCHTAB are used to evaluate f(VOL) component of outflow demand
OUTDGT	NEXITS	1	-	ft3/s	m3/s	g(t) component of outflow demand
IVOL	1	1	-	ac.ft/ ivld	Mm3/ ivld	Inflow to the RCHRES
ICON	NCONS	1	-	qty/ivld	qty/ivld	Inflow of conservative constits.
SOLRAD	1	1	-	Ly/ivld	Ly/ivld	Solar radiation
CLOUD	1	1	-	tenths	tenths	Cloud cover (range 0 thru 10)
DEWTMP	1	1	-	DegF	DegC	Dewpoint
GATMP	1	1	-	DegF	DegC	Air temp at met. station
WIND	1	1	-	miles/ ivld	km/ ivld	Wind movement

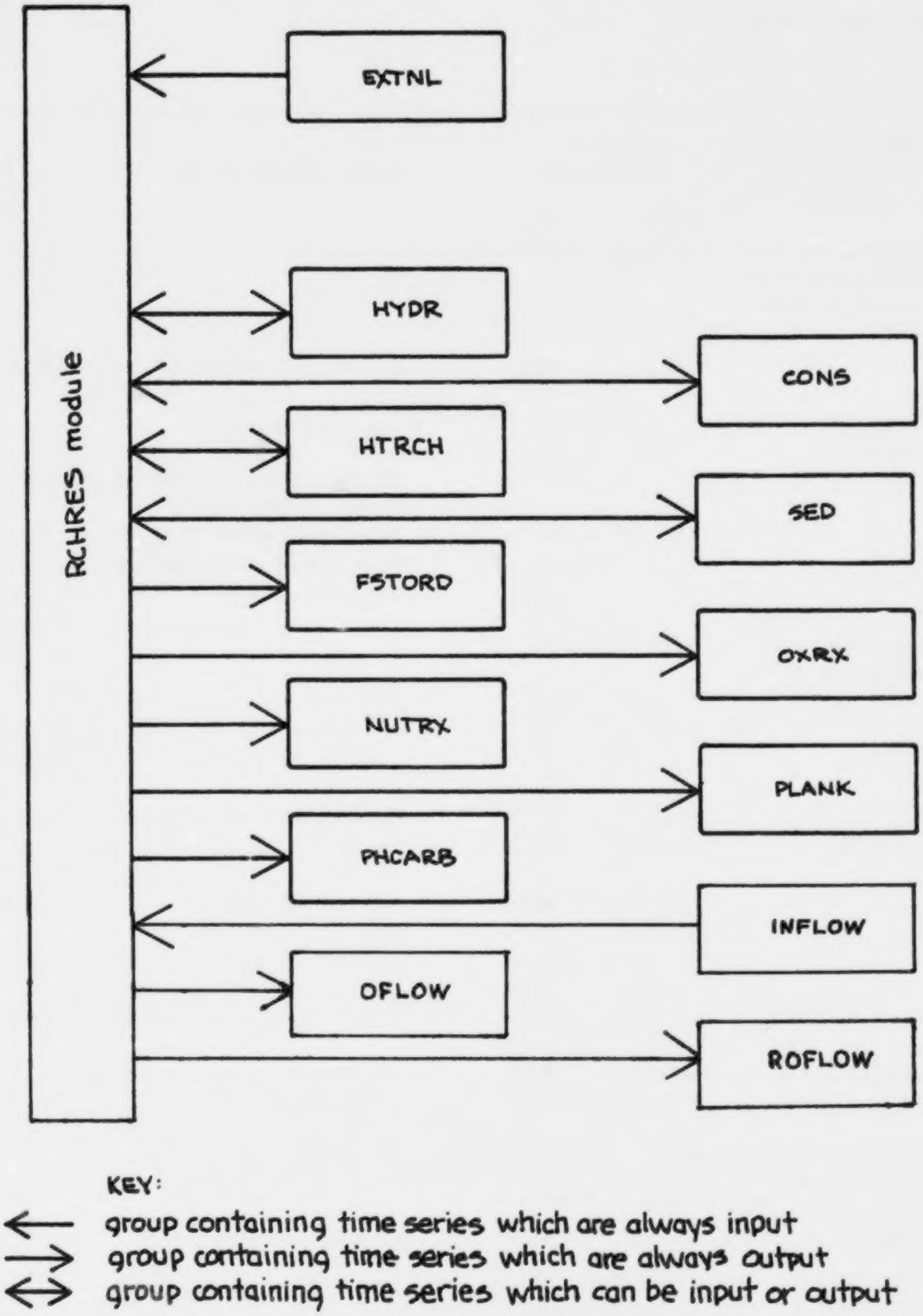

RCHRES module
EXTNL
HYDR
CONS
HTRCH
SED
FSTORD
OXRX
NUTRX
PLANK
PHCARB
INFLOW
OFLOW
ROFLOW
KEY:
← group containing time series which are always input
→ group containing time series which are always output
↔ group containing time series which can be input or output

Figure 4.7(3)-1 Groups of time series associated with the RCHRES module

4.7(3).2 Group HYDR

<---- Member ---->			Kind	Units (external)		Description/comment
Name	Max subscr values 1	2		Engl	Metr	
Time series computed by module section HYDR:						
VOL	1	1	*	ac.ft	Mm3	Volume of water in the RCHRES
AUX1FG must be 1 for next 5 members to be computed:						
DEP	1	1	*	ft	m	Depth at specified location
STAGE	1	1	*	ft	m	Stage (DEP+STCOR)
AVDEP	1	1	*	ft	m	Average depth (volume/surface area)
TWID	1	1	*	ft	m	Mean topwidth (surface area/length)
SAREA	1	1	*	ac	ha	Surface area
AUX2FG must be 1 for next 2 members to be computed:						
AVVEL	1	1	*	ft/s	m/s	Average velocity (RO/VOL)
AVSECT	1	1	*	ft2	m2	Cross-sectional area averaged over length of RCHRES (VOL/length)
RO	1	1	*	ft3/s	m3/s	Total rate of outflow from RCHRES
O	NEXITS	1	*	ft3/s	m3/s	Rates of outflow through individual exits
PRSUPY	1	1	-	ac.ft/ ivld	Mm3/ ivld	Volume of water contributed by precip on surface
VOLEV	1	1	-	ac.ft/ ivld	Mm3/ ivld	Volume of water lost by evap
ROVOL	1	1	-	ac.ft/ ivld	Mm3/ ivld	Total volume of outflow from RCHRES
OVOL	NEXITS	1	-	ac.ft/ ivld	Mm3/ ivld	Volume of outflow through individual exits
Input time series reqd to compute the above:						
Group EXTNL						always required
IVOL						
PREC						optional
POTEV						optional
COLIND						reqd only if ODFVFG is negative for one or more outflow demands
OUTDGT						reqd only if ODGTFG is >0 for one or more outflow demands

4.7(3).3 Group ADCALC

<---- Member ---->			K	Units (external)		Description/comment
Name	Max subscr values 1	2	i n d	Engl	Metr	

Time series computed by module section ADCALC:

None of the computed time series are outputtable; they are passed internally to any active "quality" sections of the RCHRES module

Input time series reqd to compute the above:

Name						Description/comment
Group HYDR						only reqd if section HYDR inactive
VOL						
O						

4.7(3).4 Group CONS

<---- Member ---->			K	Units (external)		Description/comment
Name	Max subscr values 1	2	i n d	Engl	Metr	

Time series computed by module section CONS:

Name	Max subscr values 1	2	Kind	Engl	Metr	Description/comment
CON	NCONS	1	*	concid	concid	Concentration of conservative constits
ROCON	NCONS	1	-	qty/ivld	qty/ivld	Total outflow of conservatives
OCON	NEXITS	NCONS	-	qty/ivld	qty/ivld	Outflow of conservatives through individual exits

Input time series reqd to compute the above:

Group EXTNL

ICON

4.7(3).5 Group HTRCH

<---- Member ----> Name	Max subscr values 1	 2	Kind	Units (external) Engl	 Metr	Description/comment
Time series computed by module section HTRCH:						
TW	1	1	*	DegF	DegC	Simulated water temperature
AIRTMP	1	1	*	DegF	DegC	Air temperature, adjusted for elev diff between gage and RCHRES
HTEXCH	1	1	-	BTU/ivld	kcal/ivld	Net heat exchanged with atmosphere
ROHEAT	1	1	-	"	"	Total outflow of thermal energy through active exits
OHEAT	NEXITS	1	-	"	"	Outflow of thermal energy through individual exits
Input time series reqd to compute the above:						
Group EXTNL						always required
IHEAT						optional
SOLRAD						
PREC						optional
CLOUD						
DEWTMP						
GATMP						
WIND						
Group HYDR						only reqd if section HYDR inactive
AVDEP						

4.7(3).6 Group SED

Name	<---- Member ----> Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series computed by module section SED:						
WASH	1	1	*	mg/l	mg/l	Washload concentration
SAND	1	1	*	mg/l	mg/l	Sandload concentration
BDSAND	1	1	*	ton	tonne	Bed storage of sand
SDCF1	2	2	-	ton/ivld	tonne/ivld	

In the above, the elements of array SDCF1 are:
SDCF1(1,1) deposition of washload on bed
SDCF1(2,1) total outflow of washload from RCHRES
SDCF1(1,2) exchange of sand between bed and suspended storage (scour positive)
SDCF1(2,2) total outflow of sandload from RCHRES

Name	<---- Member ----> Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
SDCF2	NEXITS	2	-	ton/ivld	tonne/ivld	Outflows of washload and sandload through individual exits

In the above, the first subscript selects the exit. The second subscript selects the constituent: 1 means washload, 2 means sandload

Name	Description/comment
Input time series reqd to compute the above:	
Group EXTNL	
IWASH	optional
ISAND	optional
Group HYDR	only reqd if section HYDR inactive
AVDEP	
AVVEL	

4.7(3).7 Group FSTORD

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series computed by module section FSTORD:						
FSTOR	NFSTOR	1	*	concid	concid	Concentrations of first order constituents
ROFST	NFSTOR	1	-	qty/ivld	qty/ivld	Total outflows of first order constits from RCHRES
FSTDEC	NFSTOR	1	-	"	"	Quantity of first order constits which decays
OFST	NEXITS	NFSTOR	-	"	"	Outflows of first order constits through individual exits

In the above, the subscript with dimension NFSTOR selects the constituent and that with dimension NEXITS selects the exit.

Input time series reqd to compute the above:

Group EXTNL
IFST(*) optional

Group HTRCH only reqd if section HTRCH inactive
TW

4.7(3).8.1 Group OXRX

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series computed by module section OXRX:						
DOX	1	1	*	mg/l	mg/l	DO concentration
BOD	1	1	*	mg/l	mg/l	BOD concentration
SATDO	1	1	*	mg/l	mg/l	Saturation DO concentration
OXCF1	2	1	-	lb/ivld	kg/ivld	Total outflows of DO (OXCF1(1,1)) and BOD (OXCF1(2,1)) from the RCHRES
OXCF2	NEXITS	2	-	"	"	Outflows of DO and BOD through individual exits

In the above, the first subscript selects the exit. The second selects the constituent: 1 means DO, 2 means BOD.

Input time series reqd to compute the above:

Group EXTNL	
IDOX	optional
IBOD	optional
WIND	only needed if LKFG=1 (lake)
Group HYDR	only reqd if section HYDR inactive
AVDEP	
AVVEL	
Group HTRCH	only reqd if section HTRCH inactive
TW	

4.7(3).8.2 Group NUTRX

Time series computed by module section NUTRX:

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
NUST	5	1	*	see below		Nutrient state variables

In the above, the first subscript selects the state variable:
1 for NO3, 2 for NH3, 3 for NO2, 4 for PO4, 5 for denitrifying bacteria.
The units of NO3 through PO4 are mg/l, denit bacteria is unitless

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
NUCF1	4	1	-	lb/ivld	kg/ivld	Total outflows of nutrients from the RCHRES

In the above, the first subscript selects the nutrient:
1 means NO3, 2 means NH3, 3 means NO2, 4 means PO4

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
NUCF2	NEXITS	4	-	lb/ivld	kg/ivld	Outflows of nutrients through individual exits

In the above, the first subscript selects the exit, the second selects the nutrient - same code as in NUCF1

Input time series reqd to compute the above:

Group EXTNL	
INO3	optional
INH3	optional
INO2	optional
IPO4	optional

Group HTRCH	only reqd if section HTRCH inactive
TW	

NOTE: Ammonia, nitrite and ortho-phosphate may, or may not, be simulated, depending on the values the user assigns to NH3FG, NO2FG and PO4FG. If a constituent is not simulated, those time series associated with it in this list should be ignored.

4.7(3).8.3 Group PLANK

Time series computed by
module section PLANK:

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
PKST3	7	1	*	mg/l	mg/l	A group of state variables

In the above, the first subscript selects the state variable:
1 for dead refractory organic N (ORN)
2 for dead refractory organic P (ORP)
3 for dead refractory organic C (ORC)
4 for total organic N (TORN)
5 for total organic P (TORP)
6 for total organic C (TORC)
7 for potential BOD (POTBOD)

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
PHYTO	1	1	*	mg/l	mg/l	Phytoplankton concentration
ZOO	1	1	*	organism/l	organism/l	Zooplankton population
BENAL	1	1	*	mg/m2	mg/m2	Benthic algae
PHYCLA	1	1	*	ug/l	ug/l	Phytoplankton as chlorophyll a
BALCLA	1	1	*	ug/m2	ug/m2	Benthic algae as chlorophyll a
PKCF1	5	1	-	lb/ivld	kg/ivld	Total outflows from the RCHRES

In the above, the first subscript selects the constituent:
1 for phytoplankton, 2 for zooplankton, 3 for ORN, 4 for ORP, 5 for ORC

Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
PKCF2	NEXITS	5	-	lb/ivld	kg/ivld	Outflows through individual exits

In the above, the first subscript selects the exit, the second selects the constituent -- same code as for PKCF1.

Input time series reqd to
compute the above:

Group EXTNL	
SOLRAD	required
IPHYTO	optional
IZOO	optional
IORN	optional
IORP	optional
IORC	optional
Group HYRCH	only reqd if section HTRCH inactive
TW	
Group SED	only reqd if section SED inactive
WASH	

NOTE: Phytoplankton, zooplankton and benthic algae may, or may not, be simulated, depending on the values the user assigns to PHYFG, ZOOFG and BALFG. If a constituent is not simulated, those time series associated with it in this list should be ignored.

4.7(3).8.4 Group PHCARB

<—— Member ——> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment

Time series computed by
module section PHCARB:

PHST	3	1	*	see below		State variables

In the above, the first subscript selects the state variable:
1 for total inorganic carbon (TIC) -- units mg/l
2 for carbon dioxide (CO2) -- units mg/l
3 for pH

PHCF1	2	1	-	lb/ivld	kg/ivld	Total outflows of TIC and CO2

In the above, the first subscript selects the constituent:
1 for TIC, 2 for CO2

PHCF2	NEXITS	2	-	lb/ivld	kg/ivld	Outflows of TIC and CO2 through individual exits

In the above, the first subscript selects the exit and the second the constituent -- same code as for PHCF1

Input time series reqd to
compute the above:

Group EXTNL	
ITIC	optional
ICO2	optional
Group CONS	only reqd if section CONS inactive
CON(ALKCON)	
Group HTRCH	only reqd if section HTRCH inactive
TW	

4.7(3).9 Groups INFLOW, ROFLOW and OFLOW

The members in these groups represent the total inflow, total outflow and outflow through individual RCHRES exits of every simulated constituent. These groups were included in the catalog to make it easier for users to specify the linkages representing time series passed from one RCHRES to another. For example, assume the RCHRES's in a run have sections HYDR, HTRCH and OXRX active, and the NETWORK Block contains:

```
NETWORK
<-Volume-> <-Grp> <-Member-><--Mult-->Tran <-Target vols> <-Grp> <-Member->  ***
<Name>   #        <Name> # #<-factor->strg <Name>   #   #        <Name> # #  ***

RCHRES   1 ROFLOW                          RCHRES   2     INFLOW
RCHRES   2 OFLOW         2                 RCHRES   3     INFLOW
```

These entries mean that the entire outflow from RCHRES 1 goes to RCHRES 2, and that the outflow through exit 2 of RCHRES 2 goes to RCHRES 3. Because the "member name" fields have been left blank, HSPF will automatically expand the above entries, generating an entry for each member which is active in this run. In this case, there will be 4 generated entries because 4 constituents are being simulated (water, heat, DO and BOD). The second set of generated entries would be:

```
NETWORK
<-Volume-> <-Grp> <-Member-><--Mult-->Tran <-Target vols> <-Grp> <-Member->  ***
<Name>   #        <Name> # #<-factor->strg <Name>   #   #        <Name> # #  ***

RCHRES   2 OFLOW  OVOL   2 1     1.0       RCHRES   3     INFLOW IVOL   1 1
RCHRES   2 OFLOW  OHEAT  2 1     1.0       RCHRES   3     INFLOW IHEAT  1 1
RCHRES   2 OFLOW  OXCF2  2 1     1.0       RCHRES   3     INFLOW OXIF   1 1
RCHRES   2 OFLOW  OXCF2  2 2     1.0       RCHRES   3     INFLOW OXIF   2 1
```

Thus, the user can specify the linkage between two RCHRES's with a single entry, instead of having to supply an entry for every constituent passed between them.

4.7(3).9.1 GROUP INFLOW

The members in this group represent the inflows to a RCHRES. Note that each member listed below is "available" for use only if the module section to which it belongs is active.

<---- Member ---->			Kind	Units (external)		Molule section	Constituent	
Name	Max subscr values 1	2		Engl	Metr			
IVOL	1	1	-	ac.ft/ ivld	Mm3/ ivld	HYDR	Water	
ICON	NCONS	1	-	qty/ ivld	qty/ ivld	CONS	Conservatives	
IHEAT	1	1	-	BTU/ ivld	kcal/ ivld	HTRCH	Heat (relative to freezing)	
IWASH	1	1	-	ton/ ivld	tonne/ ivld	SED	Washload	
ISAND	1	1	-	ton/ ivld	tonne/ ivld	SED	Sandload	
IFST	NFSTOR	1	-	qty/ ivld	qty/ ivld	FSTORD	First-order constituent	
OXIF	2	1	-	lb/ ivld	kg/ ivld	OXRX	1. DO 2. BOD	Selected using the first subscript
NUIF	4	1	-	lb/ ivld	kg/ ivld	NUTRX	1. NO3 2. NH3 3. NO2 4. PO4	
PKIF	5	1	-	lb/ ivld	kg/ ivld	PLANK	1. Phyto 2. Zoo 3. ORN 4. ORP 5. ORC	
PHIF	2	1	-	lb/ ivld	kg/ ivld	PHCARB	1. TIC 2. CO2	

4.7(3).9.2 Group ROFLOW

The members in this group represent the total outflow from a RCHRES. Note that a member is "available" for use only if the module section to which it belongs is active.

<---- Member ---->	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Module section	Constituent
ROVOL	1	1		See data for corresponding member in grp INFLOW			Water
ROCON	NCONS	1					Conservatives
ROHEAT	1	1					Heat
ROWASH	1	1					Washload
ROSAND	1	1					Sandload
ROFST	NFSTOR	1					First-order constits.
OXCF1	2	1					DO, BOD
NUCF1	4	1					NO3, NH3, NO2, PO4
PKCF1	5	1					Phyto, Zoo, ORN, ORP, ORC
PHCF1	2	1					TIC, CO2

4.7(3).9.2 Group OFLOW

The members in this group represent the outflows through the individual exits of a RCHRES. Note that a member is available for use only if the module section to which it belongs is active.

For each member, the RCHRES exit is selected by the value given to the first subscript.

<---- Member ---->			Kind	Units (external)		Module section	Constituent
Name	Max subscr values 1	2		Engl	Metr		
OVOL	NEXITS	1		See data for corresponding member in grp INFLOW			Water
OCON	NEXITS	NCONS					Conservatives
OHEAT	NEXITS	1					Heat
SDCF2	NEXITS	1					Washload
SDCF2	NEXITS	2					Sandload
OFST	NEXITS	NFSTOR					First-order constits.
OXCF2	NEXITS	2					DO, BOD
NUCF2	NEXITS	4					NO3, NH3, NO2, PO4
PKCF2	NEXITS	5					Phyto, Zoo, ORN, ORP, ORC
PHCF2	NEXITS	2					TIC, CO2

4.7(11) Catalog for COPY module

The time series groups associated with this application module are shown in Figure 4.7(11)-1.

The members contained within each group are documented in the tables which follow.

4.7(11).1 Group INPUT

<—— Member ——>			K	Units		Description/comment
	Max subscr		i	(external)		
Name	values		n			
	1	2	d	Engl	Metr	
Time series input to module COPY:						
POINT	NPT	1	*	anything		Point-valued input time series
MEAN	NMN	1	-	anything		Mean-valued input time series

4.7(11).2 Group OUTPUT

<—— Member ——>			K	Units		Description/comment
	Max subscr		i	(external)		
Name	values		n			
	1	2	d	Engl	Metr	
Time series output by module COPY:						
POINT	NPT	1	*	anything		Point-valued output time series
MEAN	NMN	1	-	anything		Mean-valued output time series
Input time series reqd to produce the above:						
Group INPUT						
POINT						reqd if NPT> 0
MEAN						reqd if NMN> 0

4.7(12) Catalog for PLTGEN module

There is only one time series group associated with this module; group INPUT, which contains all point-valued and/or mean-valued members that are to be plotted. This module does not have an output group because all its output goes to the "plot file", which is documented in Section 4.4(12) of Part E.

4.7(12).1 Group INPUT

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series input to module PLTGEN:						
POINT	NPT	1	*	anything		Point-valued input time series
MEAN	NMN	1	-	anything		Mean-valued input time series

4.7(13) Catalog for DISPLY module

There is only one time series group (INPUT) with one member (TIMSER) associated with this module since the module displays only one time series at a time. This module does not have an output group because all its output goes to the "display file" (printed).

4.7(13).1 Group INPUT

<---- Member ----> Name	Max subscr values 1	Max subscr values 2	Kind	Units (external) Engl	Units (external) Metr	Description/comment
Time series input to module DISPLY:						
TIMSER	1	1	-	anything		A mean-valued input time series

4.7(14) Catalog for DURANL module

There is only one time series group (INPUT) with one member (TIMSER) associated with this module since the module analyzes only one time series at a time. This module does not have an output group because all its output is printed. The format is documented in Section 4.2(14) of Part E.

4.7(14).1 Group INPUT

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series input to module DURANL:						
TIMSER	1	1	-	anything		A mean-valued input time series

4.7(15) Catalog for GENER module

This module has both input and output groups, like module COPY (Figure 4.7(11)-1).

The members contained within each group are documented in the tables which follow.

4.7(15).1 Group INPUT

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series input to module GENER:						
ONE	1	1	-	anything		First input time series
TWO	1	1	-	anything		Second input time series

4.7(15).2 Group OUTPUT

<---- Member ----> Name	Max subscr values 1	2	Kind	Units (external) Engl	Metr	Description/comment
Time series output by module GENER:						
TIMSER	1	1	-	anything		Output time series (mean-valued)
Input time series reqd to produce the above:						
Group INPUT						
ONE						always required
TWO						Only required if generation option needs two inputs.

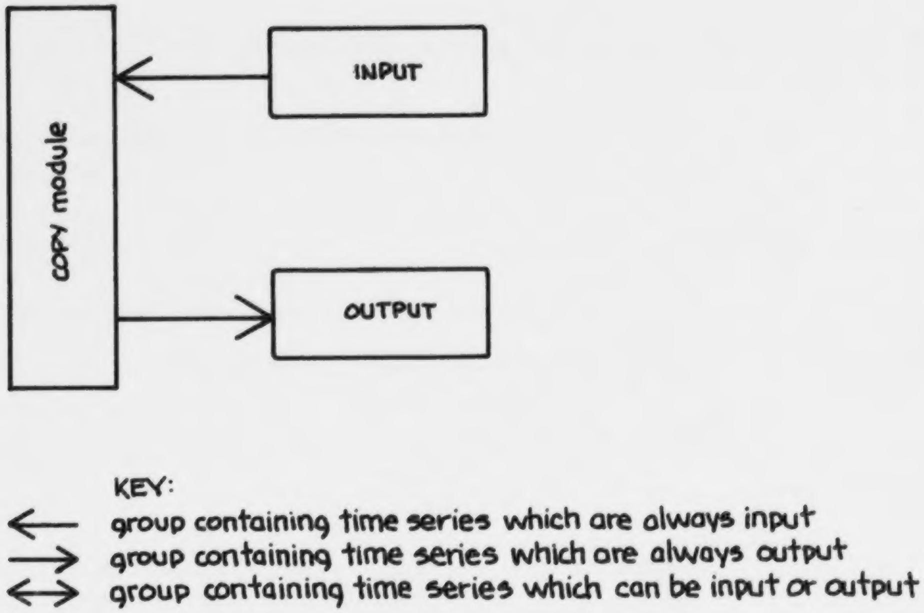

COPY module
INPUT
OUTPUT
KEY:
← group containing time series which are always input
→ group containing time series which are always output
↔ group containing time series which can be input or output

Figure 4.7(11)-1 Groups of time series associated with the COPY module

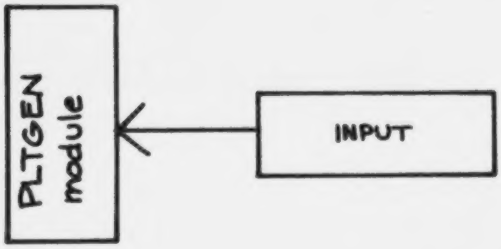


Figure 4.7(12)-1 Groups of time series associated with the PLTGEN module

4.8 FORMATS Block

Layout

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************

FORMATS
***
<ft><-------------------------- obj-fmt --------------------------------------->
.
*** line immed above repeats until all formats have been covered
.
.
END FORMATS

********************************************************************************
```

Details

Symbol	FORTRAN Name(s)	Format	Comment
<ft>	FMTCOD	I4	Identifying no. which corresponds to format no. in EXT SOURCES or TARGETS Blocks.
<obj-fmt>	FORM(19)	19A4	Standard FORTRAN object-time format.

Explanation

This block is only required if a user wishes to override the default format for reading or recording data on a sequential file (see Section 4.9).

4.9 Sequential File Formats

The following formats, for transfer of data to or from sequential files, are presently supported in the HSPF system:

4.9.1 Format class HYDFIV

It is used for the input of 5-minute data. The sequence of information is:

1. Alpha-numeric station number or identifier (this field is not read)
2. Last two digits of calendar year
3. Month
4. Day
5. Card number 1 is for midnight to 3 am.
   2 is for 3 am to 6 am.
   3 is for 6 am to 9 am.
   4 is for 9 am to noon.
   5 is for noon to 3 pm.
   6 is for 3 pm to 6 pm.
   7 is for 6 pm to 9 pm.
   8 is for 9 pm to midnight.
6. 36 fields for 5-minute data.

The default format is: (1X,3I2,I1,36F2.0)

4.9.2 Format class HYDFIF

It is used for the input of 15-minute data. The sequence of information is:

1. Alpha-numeric station number or identifier (this field is not read).
2. Last two digits of the calendar year
3. Month
4. Day
5. Card number (same as for HYDFIV above)
6. 12 fields for 15-minute data

The default format is: (1X,3I2,I1,12F6.0)

4.9.3 Format class HYDHR

It is used for input of hourly observations. The sequence of information is:

1. Alpha-numeric station no. or identifier.
   (This field is not read)
2. Last two digits of calendar year
3. Month
4. Day
5. Card no: 1 is for a.m. hours
   2 is for p.m. hours
6. Twelve fields for hourly data

The default format is:
(10X,I2,1X,I2,1X,I2,1X,I1,12F5.0)

4.9.4 Format class HYDDAY

It is used for input of daily observations. The sequence of information is:

1. Alpha-numeric station no. or identifier.
   (This field is not read)
2. Last two digits of calendar year
3. Month
4. Card no: 1 is for days 1-10
   2 is for days 11-20
   3 is for days 21-
5. Ten fields, for the daily data (11 fields for card no. 3)

The default format is:
(7X,2I2,I1,11F6.0)

4.9.5 Format class HYDSMN

It is used for input of semi-monthly observations. The sequence of information is:

1. Alpha-numeric station no. or identifier.
   (This field is not read)
2. Last two digits of calendar year
3. Card no: 1 for January through June
   2 for July through December
4. Twelve semi-monthly fields

The default format is:
(7X,I2,I1,12F5.0)

4.9.6 Format class HYDMON

It is used for input of monthly observations. The sequence of information is:

1. Alpha-numeric station no. or identifier. (This field is not read) 2. Last two digits of calendar year 3. Twelve monthly fields

The default format is: (6X,I2,12F6.0)

Note that the user can override the above default formats with his own format, supplied in the FORMATS BLOCK. He can not, however, alter the sequence of information within each record.

4.10 SPEC-ACTIONS Block

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************
Layout
******

SPEC-ACTIONS
  <-opn-range->     <--date/time---><tc><--addr--><-actcod-><-value-->
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  (repeats until all special actions have been specified)
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
END SPEC-ACTIONS

*******
Example
*******

SPEC-ACTIONS
  **************************************************************************
  Operations          Date and time  Type-      Addr   Action-   Quantity***
  Type     # to#                     code                code            ***
  **************************************************************************

  Increment surface storage of pesticide to represent field applic.      ***
  PERLND  1           1990/01/01 03     3       2511         2       80.0
  PERLND  1           1990/01/01 03     3       2514         2       80.0
  PERLND  1           1990/01/01 03     3       2517         2       80.0
END SPEC-ACTIONS

********************************************************************************
```

Details

Symbol	Fortran name(s)	Format	Def	Min	Max
<opn-range>	OPTYP(*)	3A2	none	none	none
	TOPFST	I3	see below		
	TOPLST	I4	see below		
<date/time>	DATIM(*)	I4 4(1X,I2)	see below		
<tc>	TYPCOD	I4	none	1	4
<addr>	ADDR	I10	none	1	none
<actcod>	ACTCOD	I10	none	1	2
<value>	RVAL or	F10.0	none	none	none
	IVAL	or I10	none	none	none

Explanation

<opn-range> specifies the range of operations to which this special action instruction applies (eg. PERLND 1 thru 7). The first number (TOPFST) must be greater than zero; the second (TOPLST) can be blank (zero), in which case the system assumes that only one operation is involved, else it must be >= TOPFST, to form a valid operation range. The system will perform this special action on all those operations (listed in the OPN SEQUENCE Block) which fall within <opn-range>.

The <date/time> field contains the date and time at which the special action is to be taken. The conventions regarding default values, etc. are similar to those for the <s-date-time> field in the GLOBAL Block. Thus, for example, if a year and a month are supplied but no day, hour or minute, the action will be taken at the start of the specified month.

TYPCOD indicates the "type" of the variable to be altered:

TYPCOD	Type
1	INTEGER*2
2	INTEGER*4
3	REAL*4
4	REAL*8

ADDR is the address (in COMMON Block SCRTCH) of the variable to be acted upon, expressed as a multiple of the type of variable specified by TYPCOD. The first location has the address "1" (not zero, as used in symbol maps produced by some computer systems). In the example above, the first instruction says that the 2511th location in COMMON Block SCRTCH, reckoned in multiples of REAL*4 (because TYPCOD=3), is to be altered. To find the value of ADDR appropriate to a given quantity, you should:

1. Look at the Operation Status Vector (OSV) for the module, in the data structure documentation in the Programmer's Supplement (eg. OSV for PERLND)

2. Find the Fortran name of the variable or array element that is to be acted on. For example, SPSB(3,2,1) for the storage of the first pesticide in the second "block" of the land segment, in solution form.

3. Find the location of this variable or array element in COMMON. This can be done by counting through the COMMON Block or, better, by looking at a symbol map and making any neccessary conversions to express the address using the convention outlined earlier.

ACTCOD is the "action code". 1 means reset the quantity in the location with address ADDR to <value>. 2 means increment the quantity by <value>.

The <value> field contains quantitative data for the action to be taken. If the variable or array element to be acted on is an integer (TYPCOD=1 or 2) <value> is read as an integer (IVAL); If it is REAL (TYPCOD=3 or 4), <value> is read as a real number (RVAL). Note that the value must be given in the units used internally for the quantity concerned, because no conversion is performed when it is read in. You can find the internal units by looking up the quantity in the Operations Status Vector (for the module concerned), contained in the Programmer's Supplement. For example:

1. Pesticide storage (module PERLND) has units of lb/ac (English) and kg/ha (Metric); the same units are used internally and externally.

2. Sediment storage (module PERLND) has internal units of tons/acre (in both English and Metric systems) but the external units (English and Metric) are tons/acre and tonnes/ha respectively.

For a discussion of the purpose of this Block, see Section 4.03 of Part E.

5.0 SAMPLE RUN DATA SET

Sample RUN data sets appear in Appendix II.

APPENDIX I GLOSSARY OF TERMS

1.0 NATURE OF THE GLOSSARY

The glossary which follows is not exhaustive. Its function is to introduce terms which may be new and to assign definite meanings to ambiguous terms. It is not a dictionary. The goal is not to provide formally correct definitions but to supply explanations adequate for practical purposes. Thus in some cases, the definition of a term is followed by a further explanatory note.

The definitions given in the ANSI Fortran Manual (ANSI 1966) take precedence over those given below or elsewhere when they are used in a Fortran context.

2.0 GLOSSARY

The list that follows is arranged alphabetically. Any word enclosed in parentheses ( ) may optionally be omitted from a term in everyday use, provided that the context ensures that its use is implied.

ACTUAL ARGUMENT
The name of an item (or set) of data which is being passed to (or retrieved from) a subprogram via an argument list. It can be:

(1) a variable name
(2) an array or array element name
(3) any other expression
(4) the name of an external procedure
(5) a Hollerith constant

APPLICATION MODULE
A module which simulates processes which occur in the real world.

BLOCK DATA
See ANSI Fortran Manual

BUFFER
A portion of machine memory space used for the temporary storage of input or output-bound data.

CODE
A general name for a set of statements or instructions, for example, pseudo code, Fortran code, machine code.

(CODE) SEGMENT as used in HP3000 documents
A group of program units which are loaded into machine memory together. The term may be used to refer to either source or machine code.

(CODE) SEGMENT as used in program design
See "program segment."

COMPUTATIONAL ELEMENT
See "element."

CONCEPTUAL DATA STREAM
A stream of related data that are independent of any physical input-output device.

DATA SEGMENT (especially in HP3000 documents)
The machine memory space required for data storage when executing a program. (On some systems this is not distinct from the space required by code.)

DIRECTORY
The first dataset in the time series store. It contains directory information to the other datasets.

DUMMY ARGUMENT
The local name (in a subprogram) for an actual argument which is passed to the subprogram.

ELEMENT as used in Fortran
An item in an array, for example, A(I,J).

ELEMENT as used in simulation
A group of nodes and zones, e.g. segment no. 1, reach no. 20.

ELEMENT TYPE
A name which describes elements having a common set of attributes, for example, Pervious Land-segment, Reach/Mixed Reservoir.

EXECUTABLE PROGRAM
A self contained computing procedure. It consists of a main program and its required subprograms.

(EXTERNAL) PROCEDURE
A Fortran subprogram or a procedure written in another language and called from a Fortran program.

FEEDBACK ELEMENT
An element which is situated in a loop in a network or which is connected to another element by one or more bi-directional flux linkages.

FEEDBACK REGION .
A group of connected feedback elements. Information and constituent transfers across the boundaries of a feedback region are uni-directional, but internal fluxes can be bidirectional.

FLOWCHART
A schematic two-dimensional representation of the logic in a program or program unit. The level of detail in a flowchart depends on its purpose.

FUNCTION (as used in Fortran language)
See ANSI Fortran Manual

FUNCTION as used in program design, not in Fortran language
A transformation which receives input and returns output in a predictable manner. Most functions within a program can be classified into one of three types: input, process, or output. Usually, there is a hierarchy of functions--high level functions contain subordinate functions.

IDENTIFIER
An identifying name or number in a program unit, for example, the name assigned to a subprogram, data item or COMMON block, or the number used to label a statement or to refer to an I-O unit.

INPUT TIME SERIES
Time series which are read in a given simulation run.

IVL
See SIMULATION INTERVAL

MAIN PROGRAM (UNIT)
See ANSI Fortran Manual.

MIXED RESERVOIR
A water body which is assumed to be completely mixed.

MODEL
A set of algorithms, set in a logical structure, which represents a process. A model is implemented using modules of code.

MODULE
A set of program units which performs a clearly defined function.

MODULE SECTION
A part of an Application Module which can be executed independently of the other parts. eg. SEDMNT in module PERLND.

NETWORK
A group of connected processing units. Information and/or constituents flow between processing units through uni-directional linkages. That is, no processing unit may pass output which indirectly influences itself (no feedback loops). These constraints make it possible to operate on each processing unit separately, considering them in an "upstream" to "downstream" order.

OPERATION
In HSPF: execution of code which transforms a set of input time series into a set of output time series, for example, execution of a application module or a utility module. See "simulation operation," "utility operation."

OUTPUT TIME SERIES
Time series which are generated during a simulation run. They do not have to be stored in the time series store.

PHYSICAL PROCESS
A process occuring in the real world.

PROCEDURE
See "external procedure."

PROCESS

(1) On the HP3000: An executing program together with its associated data, both internal and external.

(2) In the real world: A continuing activity, for example, percolation, chemical reaction. See "physical process."

PROGRAM
A complete set of code, consisting of one or more program units, the first of which is the "main" program unit.

PROGRAM DESIGN
The design of a set of algorithms and data structures which meets the specifications for a software system.

(PROGRAM) SEGMENT
A portion of a program which performs a single, well defined function. All parts of the function are contained within it. Note that, since a function may contain lower level functions, a segment may also contain subordinate segments.

PROGRAM UNIT
A main program or a subprogram.

PROTOTYPE
(A portion of) the real world.

PSEUDO CODE
An English-like representation of the logic in a program unit (see Textbook).

RCHRES
A Reach or Mixed Reservoir

REACH
A free-flowing portion of a stream, simulated in HSPF using storage routing.

SIMPLE ELEMENT
An element which is not a feedback element.

SIMULATION
Imitation of the behavior of a prototype, using a model. We implement the model on a computer using an application module.

SIMULATION INTERVAL
The internal time step used in an operation.

SIMULATION MODULE
See APPLICATION MODULE

SIMULATION (OPERATION)
Simulation of a specified prototype for a specified period.

SOFTWARE
A logically complete set of code and user documentation. This term is generally reserved for code which is designed to be used by others and which conforms to a standard language, with minor specified extensions.

STRUCTURE CHART
A diagram which documents the result of structured (program) design. It indicates the program units, their relationships (including hierarchy) and, optionally, the data passed between them.

STRUCTURED PROGRAM or SUBPROGRAM
A (sub)program constructed according to the principles of structured programming.

STRUCTURED (PROGRAM) DESIGN
A set of program design techniques and guidelines that are used to specify the functions in a program , the data processed by each function, and the relationships among the functions.

STRUCTURED (PROGRAM) SEGMENT
A program segment constructed according to the principles of structured programming.

STRUCTURED PROGRAMMING TECHNOLOGY
A set of related disciplines which emphasize structure in the design, implementation, and documentation of software.

STRUCTURED SOURCE CODE
Source code written in structured form. This includes the use of:

(1) indentation appropriate to the logical level of the code
(2) "intelligent" identifiers
(3) narrative commentary, to assist the reader in understanding the code.

STRUCTURE FIGURE
One of the five basic logical constructions which make up a structured program. They are commonly called:

SEQUENCE
IFTHENELSE
WHILEDO
DOUNTIL
CASE

(see Textbook)

SUBPROGRAM
The Pseudo or Fortran code used to implement a single box in a Structure chart.

SUBPROGRAM (GROUP)
The pseudo or Fortran code used to implement a box in a Structure Chart and all its subordinate functions. The group is usually referred to by the name of the top-most box.

SUBPROGRAM (UNIT)
There are three types:

subroutine
function
block data

SUBROUTINE
See ANSI Fortran Manual

SYSTEM DESIGN
The process in which program specifications, such as program requirements and record descriptions, are defined.

SYSTEM DOCUMENTATION .
A comprehensive set of documents which enable a user to understand and use a software product. It should include:

(1) a discussion of the underlying principles
(2) a discussion of the mathematical relations which the code implements
(3) documentation of the structure of the code
(4) a listing of the code
(5) documentation of data and file structures, including the input required to run the program.

TIME SERIES
A series of chronologically ordered values giving a discrete representation of the variation in time of a given entity.

(TIME SERIES) DATASET
A dataset in the time series store, containing one or more time series.

(TIME SERIES) STORE
The single direct access file used for medium/long term storage of time series. It corresponds to the OSFILE in HSPX.

TOP DOWN STRUCTURED PROGRAM or SUBPROGRAM
A structured (sub)program with:

(1) code logically segmented and implemented in a hierarchical manner, so that new code is only dependent on that which already exists.

(2) control of execution between code segments restricted to transfer between adjacent levels in the hierarchy (except for calls to service subprograms).

USER'S CONTROL INPUT
The file in which the user specifies the operations to be performed in a run, the parameters and initial conditions for each one, and the time series to be passed between them. HSPF reads this from a card image sequential file. It is also used to instruct the TSSMGR section of HSPF.

UTILITY MODULE
A module which performs operations on time series which are peripheral to the simulation of physical processes, for example, data input, plot generation, statistical analysis.

UTILITY OPERATION
Execution of a utility module.

3.0 HIERARCHY OF TERMS DESCRIBING SOURCE CODE

When discussing a program we often need to refer to a specific part or class of parts. In a structured program there is a natural hierarchial arrangement, so the various classes of parts can be readily identified and named. Using the terms defined above, the arrangement is:

- PROGRAM
  - PROGRAM UNIT (MAIN PROGRAM, SUBPROGRAM)
    - CODE SEGMENT, STRUCTURE FIGURE
      - CODE STATEMENT

The progression above is toward increasing levels of detail. Note that a "code segment" and a "structure figure" are at the same level because it is possible for either type to contain the other.

4.0 REFERENCES

International Business Machines Inc. 1974. Structured Programming Textbook & Workbook — Independent Study Program.

American National Standards Institute. 1966. USA Standard Fortran, Standard X3.9-1966. 36 pp.

APPENDIX II SAMPLE RUNS

1.0 USER'S CONTROL INPUT TO COPY TIME SERIES INTO THE TIME SERIES STORE

The following is a listing of the User's Control Input (UCI) which was used to input two time series into the Time Series Store (TSS).

The lines commencing with ! are job control statements (file specifications, etc.) required to run HSPF on a HP3000 computer. These vary from one type of machine to another.

The GLOBAL Block specifies the period for which data are being input (June 1974), and some other general control information.

The OPN SEQUENCE Block indicates that there are two COPY operations in the run, the first having a time step of 1 hour and the second 24 hours. Note that, if the time steps had been the same, the two time series could have been input in a single COPY operation; module COPY can handle up to 20 point-valued and 20 mean-valued time series in a single operation.

The COPY Block indicates that, for both COPY operations, a single mean-valued time series is being handled.

The EXT SOURCES Block specifies that:

1. The file with Fortran unit no. 21 contains hourly data (format HYDHR), in Metric units. Missing records are assumed to contain zeros (like NWS hourly precipitation cards). The multiplication factor field is blank, so defaults to 1.0. The time series goes to COPY operation no. 1 (time series group INPUT, member MEAN 1).

2. The file with Fortran unit no. 22 contains daily data (format HYDDAY) in Metric units. This time series goes to COPY operation no. 2.

The EXT TARGETS Block specifies that:

1. The output from COPY operation no. 1 (which came from sequential file 21) goes to dataset no. 25 in the TSS (member PRECIP 1) and is stored in Metric units. The access mode is ADD.

2. Similarly, the output from COPY operation no. 2 is to be stored in member PETDAT 1 of TSS dataset no. 26.

Note that the labels for the TSS datasets must have previously been created, and that the member id and unit system info (METR) supplied by the user must agree with the corresponding information in the label.

Module COPY is essentially a null operation. The work is done by module TSGET which inputs the time series (using info from EXT SOURCES Block) and module TSPUT which outputs it (using info from the EXT TARGETS Block). COPY does nothing to the time series; it is there because the system requires an operating module as a link between TSGET and TSPUT.

```
!JOB JUNE1974,RCJ.P8004
!FILE FTN01=INFOFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN02;REC=-84,3,F,ASCII
!FILE FTN03=ERRFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN04=WARNFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN05=$STDIN
!FILE FTN07;REC=-22,11,F,BINARY
!FILE FTN08;REC=1000,1,F,BINARY
!FILE FTN09;REC=300,1,F,BINARY
!FILE FTN10;REC=300,1,F,BINARY
!FILE FTN12;REC=60,6,F,BINARY
!FILE FTN15=TSSFL.RCJ.P8004,OLD
!FILE FTN21=PRECDATA,OLD
!FILE FTN22=PETDAT,OLD
!SETDUMP
!RUN HSPFPF.USLS.P7614
RUN

GLOBAL
  Inputting test data to TSS
  START       1974/06          END    1974/06
  RUN INTERP OUTPUT LEVEL    3
  RESUME     0 RUN     1
END GLOBAL

OPN SEQUENCE
      COPY          1   INDELT 01:00
      COPY          2   INDELT 24:00
END OPN SEQUENCE

COPY
  TIMESERIES
    #thru#  NPT  NMN ***
    1    2         1
  END TIMESERIES
END COPY

EXT SOURCES
<-Volume-> <Srcfmt> SsysSgap<--Mult-->Tran <-Target vols> <-Grp> <-Member->  ***
<Name>   #          tem strg<-factor->strg <Name>   #   #        <Name> # #  ***

SEQ     21 HYDHR    METRZERO               COPY     1     INPUT  MEAN   1
SEQ     22 HYDDAY   METRZERO               COPY     2     INPUT  MEAN   1
END EXT SOURCES
```

```
EXT TARGETS
<-Volume-> <-Grp> <-Member-><--Mult-->Tran <-Volume-> <Member> Tsys Tgap Amd ***
<Name>   #        <Name> # #<-factor->strg <Name>   # <Name> #  tem strg strg***
COPY     1 OUTPUT MEAN   1                 TSS     25 PRECIP 1 METR      ADD
COPY     2 OUTPUT MEAN   1                 TSS     26 PETDAT 1 METR      ADD
END EXT TARGETS

END RUN
!EOD
!EOJ
```

2.0 USER'S CONTROL INPUT TO SIMULATE PWATER, SEDMNT, MSTLAY AND PEST SECTIONS OF THE PERLND APPLICATION MODULE

The following is a listing of the User's Control Input (UCI) to run the PERLND Application Module, with the PWATER, SEDMNT, MSTLAY and PEST sections active. Features of the input which are common to those discussed in the previous example will not be described again.

The OPN SEQUENCE Block indicates that a single Pervious Land-segment (PLS) is being simulated.

The PERLND Block specifies the parameters and initial conditions. Table ACTIVITY indicates which module sections are active. It is possible to have any meaningful set of the sections active. Table PRINT-INFO indicates that printout will be produced every interval, because the print level flag for every section is set to 2 and PIVL = 1.

Table GEN-INFO specifies that Land-segment CRUD is to be subdivided into 3 areal source "blocks", that all inputs are in the English unit system (unit flags= 1), that printout in the English system is to go to Fortran file no. 6, and that there is not to be any printout in the Metric system.

The parameters for section PWATER appear in Tables PWAT-PARM1 through PWAT-PARM4. The initial conditions are contained in Tables PWAT-STATE1 and PWAT-BLKSTAT. Note that three of the latter are required because the land-segment is divided into three blocks.

The parameters for section SEDMNT are specified in Tables SED-PARM2 and SED-PARM3.

If the user does not supply an expected table, or if he leaves one or more fields blank in a table, HSPF supplies default values (where they exist). Thus, in this example, NSUR (in Table PWAT-PARM4), all values in Tables SED-PARM1 and SED-STOR, and all the MSTLAY parameters and initial conditions will be defaulted.

Table PEST-FLAGS indicates that only one pesticide is being simulated (NPST= 1) and that adsorption/desorption will be simulated using first-order kinetics (ADOPFG=1). Tables SOIL-DATA through PEST-DEGRAD specify pesticide related parameters; Table PEST-FIRSTPM is repeated for each of the four soil layers that HSPF simulates (surface through groundwater). The initial storages of pesticide appear in Table-type PEST-STOR1. A table of this type is expected for every soil layer and, for the surface and upper layers, for every areal source block. Because tables of this type for the lower and groundwater layers are absent, initial storages in these layers will default to zero.

The EXT SOURCES Block indicates that all the required input time series are to come from the TSS. Note that the same time series is being used to supply the surface, upper and lower/groundwater layer temperatures. Thus, temperatures throughout the soil profile will be the same. To get a more realistic soil temperature distribution, the user could have specified separate sources for each of the temperature time series (possibly obtained from observed data). Alternatively, he could have activated section PSTEMP of this module, to compute the time series. In that case, the temperature data would automatically be passed to section PEST. And, because the data were computed, there would be no need to specify them as inputs in the EXT SOURCES Block; in fact, it would be an error to do so.

The value SAME (in the EXT SOURCES Block) means that, if the temperature data in the TSS have a larger time interval (say, day) than the simulation interval (30 mins), the disaggregated data will have the same values for an entire day. The default functional in such a case is DIV, which would cause the daily values to be divided into 48 equal 30-minute values; this is appropriate for data like potential ET but not for temperatures!

Note that there is no NETWORK or EXT TARGETS Block because there is only one operation in the run and no time series are being written to any external file.

```
!JOB PERLTEST,RCJ.P8004
!FILE FTN01=INFOFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN02;REC=-84,3,F,ASCII
!FILE FTN03=ERRFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN04=WARNFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN05=$STDIN
!FILE FTN07;REC=-22,11,F,BINARY
!FILE FTN08;REC=1000,1,F,BINARY
!FILE FTN09;REC=300,1,F,BINARY
!FILE FTN10;REC=300,1,F,BINARY
!FILE FTN12;REC=60,6,F,BINARY
!FILE FTN15=TSSFL.RCJ.P8004,OLD
!SETDUMP
!RUN HSPFPF.USLS.P7614
RUN

GLOBAL
  Testing sections PWATER, SEDMNT, MSTLAY and PEST with FSTORD option & NBLKS=3
  START       1990/01/01 00:00  END    1990/01/01 06:00
  RUN INTERP OUTPUT LEVEL    3
  RESUME     0 RUN     1
END GLOBAL
```

```
OPN SEQUENCE
      PERLND       1   INDELT 00:30
END OPN SEQUENCE

PERLND
  ACTIVITY
    #thru# ATMP SNOW PWAT  SED PSTP PWTG PQAL MSTL PEST NITR PHOS TRAC ***
    1    5    0    0    1    1    0    0    0    1    1
  END ACTIVITY
  PRINT-INFO
    #thru#  <--------------print level FLAGS ------------------------> PIVL ***
    1    5    2    2    2    2    2    2    2    2    2    2    2    2    1
  END PRINT-INFO
  GEN-INFO
    #thru#      Name          NBLKS   Unit systems Printout files ***
    1     Land-segment CRUD      3    1    1    1    6    0
  END GEN-INFO

  TABLES FOR SECTION PWATER ***

  PWAT-PARM1
    #THRU# CSNO RTOP ***
    1    5    0    1
  END PWAT-PARM1
  PWAT-PARM2
    #THRU#*** FOREST      LZSN    INFILT      LSUR     SLSUR     KVARY     AGWRC
    1          0.5        5.0      .30        400.       .01      .00       .98
  END PWAT-PARM2
  PWAT-PARM4
    #THRU#***  CEPSC      UZSN      NSUR     INTFW       IRC     LZETP
    1                     1.5                 2.0        .50      .50
  END PWAT-PARM4
  PWAT-STATE1
    #THRU#***    CEPS      SURS       UZS      IFWS       LZS      AGWS     GWVS
    1    5                            2.0       6.0      10.0      30.0
  END PWAT-STATE1

  PWAT-BLKSTAT      Block 1
    #thru#   Surface     Upper      Iflw ***
    1    5       0.0       2.0      6.0
  END PWAT-BLKSTAT
  PWAT-BLKSTAT      Block 2
    #thru#   Surface     Upper      Iflw ***
    1    5       0.0       2.0      6.0
  END PWAT-BLKSTAT
  PWAT-BLKSTAT      Block 3
    #thru#   Surface     Upper      Iflw ***
    1    5       0.0       2.0      6.0
  END PWAT-BLKSTAT
```

```
*** Values in Table-type SED-PARM1 defaulted
SED-PARM2
  #thru#      SMPF      KRER      JRER     AFFIX     COVER      NVSI***
  1                      5.0       2.0       .01       0.5
END SED-PARM2
SED-PARM3
  #thru#      KSER      JSER      KGER      JGER ***
  1            3.0       2.0       0.0       1.0
END SED-PARM3

*** All MSTLAY parameters and initial conditions set to default values

*** Pesticide ***

PEST-FLAGS
  #thru# NPST<-----ITMXPS--><----ADOPFG---> ***
                 1    2    3    1    2    3 ***
  1    5    1                   1
END PEST-FLAGS

SOIL-DATA
  #thru#    Depth of each layer (in)      ***
         Surface   Upper   Lower  Ground  ***
  1    5    .125     6.0    72.0    72.0
END SOIL-DATA

PEST-ID
  #thru#<--- Pesticide ----> ***
  1    5 Ant-killer #1
END PEST-ID

PEST-FIRSTPM          Surface layer
  #thru#     KDSPS     KADPS ***
  1    5       1.0       1.0
END PEST-FIRSTPM

PEST-FIRSTPM          Upper layer
  #thru#     KDSPS     KADPS ***
  1    5       1.0       1.0
END PEST-FIRSTPM

PEST-FIRSTPM          Lower layer
  #thru#     KDSPS     KADPS ***
  1    5       1.0       1.0
END PEST-FIRSTPM

PEST-FIRSTPM          Groundwater layer
  #thru#     KDSPS     KADPS ***
  1    5       1.0       1.0
END PEST-FIRSTPM
```

```
  PEST-DEGRAD
    #thru#   Surface     Upper     Lower    Ground ***
    1    5     0.01       0.01      0.01      0.01
  END PEST-DEGRAD

  Block 1 ***
  PEST-STOR1        Surface Layer
    #thru#     Cryst        Ads       Soln  (lb/ac) ***
    1    5      10.0
  END PEST-STOR1

  PEST-STOR1        Upper Layer
    #thru#     Cryst        Ads       Soln  (lb/ac) ***
    1    5                 100.0
  END PEST-STOR1

  Block 2 ***
  PEST-STOR1        Surface Layer
    #thru#     Cryst        Ads       Soln  (lb/ac) ***
    1    5      10.0
  END PEST-STOR1

  PEST-STOR1        Upper Layer
    #thru#     Cryst        Ads       Soln  (lb/ac) ***
    1    5                 100.0
  END PEST-STOR1

  Block 3 ***
  PEST-STOR1        Surface Layer
    #thru#     Cryst        Ads       Soln  (lb/ac) ***
    1    5      10.0
  END PEST-STOR1

  PEST-STOR1        Upper Layer
    #thru#     Cryst        Ads       Soln  (lb/ac) ***
    1    5                 100.0
  END PEST-STOR1

END PERLND
```

```
EXT SOURCES
<-Volume-> <Member> SsysSgap<--Mult-->Tran <-Target vols> <-Grp> <-Member->    ***
<Name>   # <Name> # tem strg<-factor->strg <Name>   #   #        <Name> # #    ***

TSS     11   PREC   ENGL                   PERLND   1     EXTNL  PREC
TSS     13 PETNUL   ENGL                   PERLND   1     EXTNL  PETINP
TSS     11   TEMP   ENGL              SAME PERLND   1     PSTEMP SLTMP
TSS     11   TEMP   ENGL              SAME PERLND   1     PSTEMP ULTMP
TSS     11   TEMP   ENGL              SAME PERLND   1     PSTEMP LGTMP

END EXT SOURCES


END RUN
!EOD
!EOJ
```

3.0 USER'S CONTROL INPUT TO SIMULATE A SET OF PERVIOUS LAND-SEGMENTS, FREE-FLOWING REACHES AND A RESERVOIR

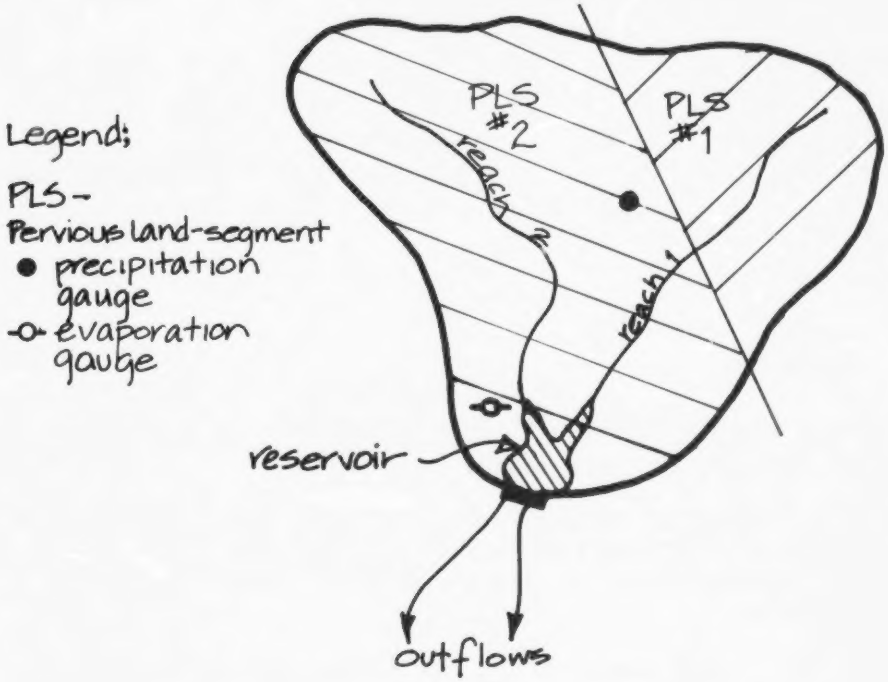
The following is a listing of the User's Control Input (UCI) to simulate two pervious land-segments (PLS 1 & 2) which feed into two free-flowing reaches (RCHRES 1 & 2) which then discharge into a reservoir (RCHRES 3). The setup is diagrammed in Figure 1.

The OPN SEQUENCE Block specifies that the two PERLND operations, three RCHRES operations, and a PLTGEN operation, are all in the same INGRP. Thus, they have the same internal time step (30 min) and share the same internal scratch pad. All time series transfers between these operations can therefore take place internally (that is, without going through the TSS), as specified in the NETWORK Block.

Table ACTIVITY, in both the PERLND and RCHRES Blocks, indicates that only water movement is being simulated; no quality constituents are considered in this example. Table PRINT-INFO, also present in both the PERLND and RCHRES Blocks, specifies a print level of 4; that is, printout will occur monthly. Table-type GEN-INFO contains the names of the PLSs and the RCHRESs and, in the latter case, also specifies the number of "exits" (outflows). RCHRES 3 has two exits; one supplies an irrigation demand, the other is for any flow over the spillway.

Tables PWAT-PARM2 and PWAT-PARM4 show that the two PLSs have different parameter sets. However, initial conditions are the same for both (Table PWAT-STATE1).

Table HYDR-PARM1 indicates that, for all three RCHRESs, the outflow demands are functions of stored volume only (ODFVFG >0, ODGTFG =0). Also, for RCHRES 3, flag A1 is ON, which means that auxiliary variables such as depth and surface area will be computed. This is required to calculate the quantity of water supplied to the reservoir by precipitation on the surface and the quantity lost by evaporation (depth*surface area). Table HYDR-PARM2



Reservoir details:

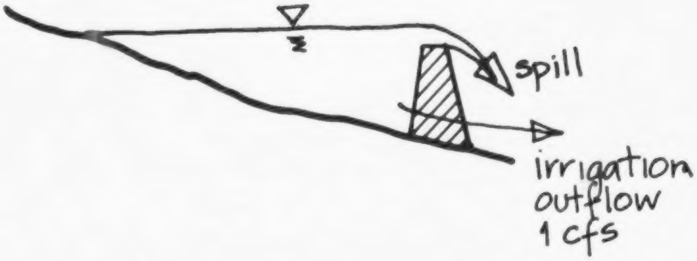


FIGURE 1 System simulated in Example in Section 3.0

indicates that RCHRESs 1 and 2 share a common FTABLE; that is, they have identical hydraulic properties.

The FTABLES Block contains the function tables, which specify the hydraulic properties of each RCHRES. Note that FTABLE 101 has two "discharge" columns, because RCHRES 3 has two exits.

The PLTGEN Block contains the data necessary to generate a "plot file", which will be read by a separate program to plot the inflow and outflow rates for the reservoir. Table PLOTINFO indicates that the plot file is Fortran unit no. 30 and that two point-valued time series are to be plotted (no mean-valued time series). Other tables give the plot labels, scaling information, etc.

The EXT SOURCES Block indicates that member PRECIP of TSS dataset no. 25 is to supply precipitation input for both PLSs and the reservoir, using a different multiplication factor each time. These operations also share the same source of potential ET data (TSS 26).

The NETWORK Block specifies the connectivity of the operations. The outflow from PLS 1 (member PERO of time series group PWATER) becomes inflow to RCHRES 1 (Figure 1). The multiplication factor converts the outflow, in inches/interval, to RCHRES inflow, in acre-ft/interval, and thus includes the tributary area (in acres) and a factor of 12 to change runoff depth from inches to feet. RCHRESs 1 and 2 both flow into RCHRES 3. The outflow rates from RCHRES 1 and 2 are both routed to member POINT 1 of group INPUT for the plot operation. Thus, the sum of these rates will be the first plotted time series. The second plotted time series will be the total outflow rate for RCHRES 3.

```
!JOB COMPRTST,RCJ.P8004
!FILE FTN01=INFOFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN02;REC=-84,3,F,ASCII
!FILE FTN03=ERRFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN04=WARNFL.RCJ.P7614,OLD;ACC=IN
!FILE FTN05=$STDIN
!FILE FTN07;REC=-22,11,F,BINARY
!FILE FTN08;REC=1000,1,F,BINARY
!FILE FTN09;REC=300,1,F,BINARY
!FILE FTN10;REC=300,1,F,BINARY
!FILE FTN12;REC=60,6,F,BINARY
!FILE FTN15=TSSFL.RCJ.P8004,OLD
!PURGE PLOTFL
!BUILD PLOTFL;REC=-80,3,F,ASCII;DISC=1500
!FILE FTN30=PLOTFL,OLD
!SETDUMP
!RUN HSPFPF.USLS.P7614
RUN
```

```
GLOBAL
  Running a network of 2 Land-segments, 2 Rchres's and 1 Plot Operation
  START       1974/06          END    1974/06
  RUN INTERP OUTPUT LEVEL    3
  RESUME     0 RUN     1
END GLOBAL

OPN SEQUENCE
    INGRP              INDELT 00:30
      PERLND       1
      PERLND       2
      RCHRES       1
      RCHRES       2
      RCHRES       3
      PLTGEN       1
    END INGRP
END OPN SEQUENCE

PERLND
  ACTIVITY
    #thru# ATMP SNOW PWAT  SED PSTP PWTG PQAL MSTL PEST NITR PHOS TRAC ***
    1    2    0    0    1
  END ACTIVITY

  PRINT-INFO
    #thru#         PFLAG ***
    1    2            4
  END PRINT-INFO

  GEN-INFO
    #thru#      Name           NBLKS                 Printout  ***
    1      Land-segment One      1                     6
    2      Land-segment Two      1                     6
  END GEN-INFO

  TABLES FOR SECTION PWATER ***

  PWAT-PARM1
    #THRU# CSNO RTOP ***
    1    2    0    1
  END PWAT-PARM1

  PWAT-PARM2
    #THRU#*** FOREST      LZSN     INFILT      LSUR     SLSUR     KVARY     AGWRC
    1          0.0       16.0       .50       300.        .24       .00       .99
    2          0.0       20.0       .45       400.        .10       .00      .995
  END PWAT-PARM2

  PWAT-PARM4
    #THRU#***  CEPSC      UZSN       NSUR     INTFW      IRC     LZETP
    1            .25      0.8         .35     3.0        .75      .30
    2            .25      1.0         .35     4.0        .85      .45
  END PWAT-PARM4
  PWAT-STATE1
    #THRU#***     CEPS      SURS       UZS      IFWS       LZS      AGWS      GWVS
    1    2                             1.2                24.5       8.5       2.0
  END PWAT-STATE1

END PERLND


RCHRES
  ACTIVITY
    #thru# HYDR ADCA CONS HEAT  SED  FST OXRX NUTR PLNK   PH ***
    1    3    1
  END ACTIVITY

  PRINT-INFO
    #thru#PFLAG ***
    1    3    4
  END PRINT-INFO

  GEN-INFO
    #thru#      Name         NEXITS    Unit flags     Print-files ***
    1      Reach 1                1    1    1    1    6    0
    2      Reach 2                1    1    1    1    6    0
    3      Reach 3(reservoir)     2    1    1    1    6    0
  END GEN-INFO

  HYDR-PARM1
    #thru#            Flags                                     ***
             V  A  A         ODFVFG             ODGTFG          ***
           CON  1  2        1  2  3  4  5      1  2  3  4  5    ***
    1    2                  4
    3           1           4  5
  END HYDR-PARM1

  HYDR-PARM2
    #thru#    FTABNO       LEN ***
    1    2       100       1.0
    3            101       0.1
  END HYDR-PARM2

  HYDR-INIT
    #THRU#      VOL(ac-ft) ***
    3          200.
  END HYDR-INIT

END RCHRES
```

```
FTABLES

  FTABLE    100
  Rows/cols ***
    5    4
     DEPTH      SAREA       VOL      Disch ***
        ft      acres   acre-ft      ft3/s ***
       0.0        0.0       0.0        0.0
       2.0       1.72       3.0       16.0
       6.0       4.44      12.0      106.0
      10.0       14.5      51.0      521.0
      15.0       40.0     200.0     2500.0
  END FTABLE100

  FTABLE    101
  Rows/cols ***
    5    5
     DEPTH      SAREA       VOL      Disch      Disch ***
        ft      acres   acre-ft      ft3/s      ft3/s ***
       0.0        0.0       0.0        1.0        0.0
      10.0        7.5       25.        1.0        0.0
      20.0       30.0      200.        1.0        0.0
      25.0        47.      390.        1.0       100.
      30.0       67.5      675.        1.0      1000.
  END FTABLE101

END FTABLES
```

```
PLTGEN
  PLOTINFO
    #thru# FILE  NPT  NMN Labl ***
    1        30    2
  END PLOTINFO

  GEN-LABELS
    #thru#<----------------TITLE ---------------->***      <------Y AXIS------->
    1     Plot of reservoir flowrates                       Flow (ft3/sec)
  END GEN-LABELS

  SCALING
    #thru#      YMIN      YMAX     IVLIN ***
    1                     300.       48.
  END SCALING

  CURV-DATA   (first curve)
    #thru#    < Curve label  > ***
    1         Inflow
  END CURV-DATA

  CURV-DATA   (second curve)
    #thru#    < Curve label  > ***
    1         Outflow
  END CURV-DATA
END PLTGEN

EXT SOURCES
<-Volume-> <Member> SsysSgap<--Mult-->Tran <-Target vols> <-Grp> <-Member->  ***
<Name>   # <Name> # tem strg<-factor->strg <Name>   #   #        <Name> # #  ***
TSS     25 PRECIP   METR          1.3      PERLND   1      EXTNL  PREC
TSS     25 PRECIP   METR          1.5      PERLND   2      EXTNL  PREC
TSS     25 PRECIP   METR          1.0      RCHRES   3      EXTNL  PREC
TSS     26 PETDAT   METR                   PERLND   1   2  EXTNL  PETINP
TSS     26 PETDAT   METR                   RCHRES   3      EXTNL  POTEV
END EXT SOURCES

NETWORK
<-Volume-> <-Grp> <-Member-><--Mult-->Tran <-Target vols> <-Grp> <-Member->  ***
<Name>   #        <Name> # #<-factor->strg <Name>   #   #        <Name> # #  ***
PERLND   1 PWATER PERO           53.3      RCHRES   1      EXTNL  IVOL
PERLND   2 PWATER PERO           36.7      RCHRES   1      EXTNL  IVOL
PERLND   2 PWATER PERO           70.0      RCHRES   2      EXTNL  IVOL
RCHRES   1 HYDR   ROVOL                    RCHRES   3      EXTNL  IVOL
RCHRES   2 HYDR   ROVOL                    RCHRES   3      EXTNL  IVOL
RCHRES   1 HYDR   RO                       PLTGEN   1      INPUT  POINT  1
RCHRES   2 HYDR   RO                       PLTGEN   1      INPUT  POINT  1
RCHRES   3 HYDR   RO                       PLTGEN   1      INPUT  POINT  2
END NETWORK

END RUN
!EOD
!EOJ
```

APPENDIX III PROGRAM NEWTSS

1.0 GENERAL DISCUSSION

1.1 Purpose

NEWTSS is a stand-alone program which creates a Time Series Store (TSS). It must, therefore , be run before a user can perform any HSPF runs which require data to be stored in, or retrieved from, the TSS. When running NEWTSS the user specifies the size of the TSS, the record length, etc. NEWTSS can also be used to copy the contents of one TSS to another. This option is used if the existing TSS is too large or too small; the user creates a new TSS of the desired size and copies everything to it.

1.2 Introduction

The Time Series Store (TSS) is a large random access file which represents time series in a convenient manner for storage and retrieval, using subroutines provided by the HSPF system. The following objectives were followed in the design of the TSS and its associated subroutines:

1. Enable uniform storage and retrieval of time series
2. Provide for suppression of redundant information
3. Provide flexibility in access methods
4. Allow grouping time series into related sets for easier reference
5. Provide for useful transformations of time series so that time series with disparate characteristics can be used by an application module in HSPF

1.3 TSS Structure

Logically, the TSS should be viewed as a large two dimensional array of REAL values (in the sense of FORTRAN) in which each row is a record. These rows or TSS records are numbered consecutively with the first record given the number 1. For our purposes a record is a collection of related items or data treated as a unit. A set of related records may form a file or a dataset. Records can be related in various ways so that a record may be associated with several groups. For example, each row of the array taken to represent the TSS is a record in the file defined for the TSS. However, the TSS is itself subdivided into groups of TSS records related by criteria other than for the TSS as whole. These subgroups will be referred to as datasets. The three types of datasets are: TSS Descriptor, TSS Directory and Time Series dataset.

1.4 TSS Descriptor

The TSS Descriptor contains information to define the characteristics of the TSS. Certain of these characteristics will vary from one TSS to another and it is therefore necessary that the TSS be self descriptive. The TSS Descriptor is contained within one TSS record and is the first record of the TSS. The length of the TSS descriptor gives the minimum length for any TSS record. Table 1 gives a description of the contents of the TSS Descriptor.

1.5 TSS Directory

The TSS directory describes itself and the time series datasets in the TSS. The directory contains a record for itself and for each time series dataset in the TSS. Each TSS directory record contains descriptive information necessary to manipulate datasets in the TSS. Table 2 gives a description of the contents of a record in the TSS directory.

The number of TSS records to allocate to the directory is set by the user when the file for the TSS is first created. This first estimate may need to be changed as the TSS is used. Thus, for the purposes of dataset manipulation, the TSS directory is not distinguished from a time series dataset.

1.6 Time Series Dataset

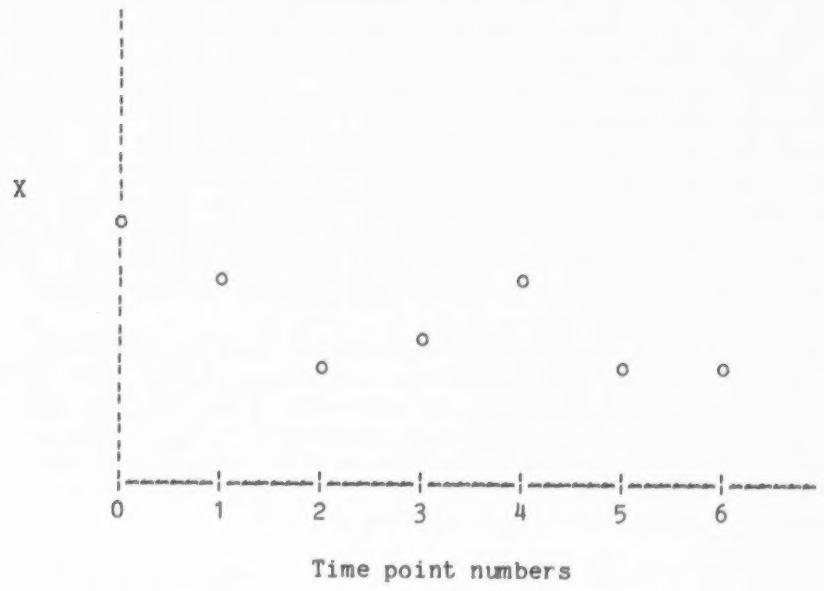
The time series dataset is the heart of the TSS and the purpose for its creation. The form of the time series and the various concepts needed for describing the time series and the data structures must be defined first. We can then describe the structure of a time series dataset in a concise manner.

1.7 Time Series Concepts

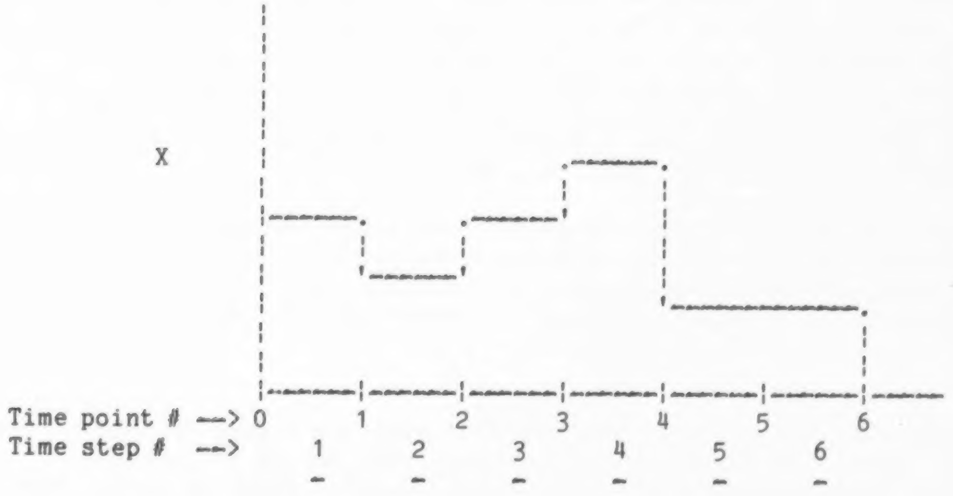
A time series is a sequence of values ordered in time. We take "value" to include vectors as well as scalars. The interval of time between successive values is called the time step or the time increment or the time interval of the time series. The time step for a time series is often a constant value but may also be variable. The implementation in HSPF restricts the variability in a manner discussed below. The values in the time series may represent the behavior of a process at a point in time or an average over the time step of the time series. A time series whose values represent behavior at points in time is called a point-valued time series and is represented symbolically by "*". Linear interpolation is used to define intermediate values in a point-valued time series. A time series whose values represent average or aggregated behavior over the time step are called mean-valued time series and are represented symbolically as "-". The meaning of "average" and "mean" is taken in a wide sense and includes any value assumed to be representative of behavior of the time series over the time step, rather than at a specific point in time.

The following figure shows the difference between the point and mean value time series in graphic form. It is important to note that only one value is needed to represent the behavior of a mean-valued time series for one time step. We visualize the value as being assigned to a time step in this case. On the other hand, two values are needed to represent the behavior of a point-valued time series over the same interval. We visualize the values as being assigned to the time points in this case. Each time point at which a value of the series is given in a point-valued time series is viewed as "belonging" to the time step which it ends. Time points belonging to all time steps contained within a larger time step are viewed as belonging to the larger time step also. For example, all time points in a point-valued time series except the first time point belong to the time interval spanning the time series duration. The first time point of a point-valued time series is viewed as belonging to the time step immediately preceding the first time step of the time series. This precise definition of belongingness for a time point is needed to avoid confusion in defining operations on the time series.

A number of operations on time series, discussed in Section 4.6 of Part F, preserve the integral of the time series between any two time points which end time steps in the time series. The integral may be visualized as the area under the broken line graph formed by connecting adjacent values in the point-valued time series or the area under the histogram representing the mean-valued time series. The trapezoidal rule applied to the point-valued time series yields the exact value of the integral whereas the simple rectangular rule yields the exact value for the mean-valued time series.



Point-valued time series



Mean-valued time series

Figure 1. Comparison between point- and mean-valued time series

The inclusion of vector valued time series requires some definitions. The adjective "time series" is used interchangably to refer to to a vector valued time series, a scalar valued time series or to a single component of a vector valued time series. Thus the word "time series" must be interpreted in context. The components in a vector valued time series are ollected into subgroups called members. The components making up each member must all be of the same kind (either point-valued or mean-valued) but the members in the time series may be of differing kinds. The rules for grouping component time series into members and the rules for grouping members into a time series dataset are not given by HSPF. The user can group them in any meaningful manner with the restriction that there be no more than 20 components in any vector valued time series dataset.

Each component of a vector valued time series must be recorded at the same time point. The time points and/or the intervals in the time series must be labelled uniquely. Unfortunately, conventional time keeping does not produce unique labels nor are the conventions universally used. Therefore, the HSPF system defines an internal time keeping system for its own use. This system has a resolution of one minute. Thus, the minimum time step is one minute and fractions of a minute cannot be represented.

Time is given as year/month/day/hour/minute to completely specify either a time interval or a time point. The date/time given by the internal clock uses the "contained within" principle for all levels of the date/time. That is, each smaller interval is contained within the next larger interval. This is the conventional usage for year/month/day but is not conventional for the hour/minute. For example, the date string 1977/01/02 labels the second day of the first month of the 1977th year. On the other hand, in conventional usage the time string 10:15 refers to the end of the 15th minute after (not within) the 10th hour of the day. This change in meaning is eliminated in the internal date/time clock for HSPF. In the internal system, time string 10/15 labels the 15th minute of (ie. within) the 10th hour of the day. A comparable time to 10:15 in the conventional sense would be 11/15; that is, the 15th minute of the 11th hour of the day.

In summary, the internal clock convention labels time intervals at all levels of date/time whereas conventional usage labels time intervals for year/month/day but labels time points for hour/minute. In HSPF, time points are then referenced uniquely by the minute which ends at the time point in question.

The time steps in a time series are labelled with the minute which ends the time step. Thus, the values in a mean-valued time series are treated logically as having occurred at the end time point of the time step. Note that for purposes of the internal clock and for description of internal concepts each time point has one and only one label. This means that we refer to the instant in time forming the boundary between two days using the label associated with the first day even though our interest is centered on the second day. This convention is called the ending time convention. A starting time convention is used externally for some purposes because traditional usage requires both conventions depending on the context of the statment about time. Users are more comfortable using the traditional clock and both a starting time and an ending time convention. The starting time convention is used when the start of some time span is in mind and the ending time convention is used when the end of some time span is in mind.

The time span associated with a time series must be defined. Logically, a time series is of infinite length. Realistically, every time series has a finite length and may be broken into short segments for convenience in recording the values on some medium such as the printed page, a magnetic tape, a data card or a magnetic disk. These shorter segments are made neccesary by various software and hardware constraints. Therefore, a time span is associated with each medium used to record or store the time series.

A further practical complication is created by the variety of representations used for time series. The user's most likely mental image is a line drawn in some coordinate system on the printed page. This method of representing time series is most convenient for the user but a series of discrete numbers is most convenient for the digital computer. The time series of indefinite length must be subdivided into shorter time spans to fit the card images or the records on the tape or disk. In some cases data for the time series may be incomplete (some values not present) or, in some cases, many of the values are zero so that not all values for the time series are stored on the medium. In such cases a date/time indicator is given on the record. As an example, think of the format used for data cards punched by the National Weather Records Center. The date/time information on each record of the medium permits the reconstruction of the complete time series (except for the missing values) even though not all values are recorded on the medium. However, conventions must be established so that missing records on a given recording medium are properly interpretted. For example, are the missing data merely zeros or did they occur because of instrument malfunction? If the data are missing, a "filler" should be inserted when the data are placed on the TSS so that it can be changed at a later time or so that such missing periods can be properly handled by other parts of the HSPF system. The filler value is called an "undefined" value in the TSS system.

The time step for a time series can vary in multiples of a basic time step established for the time series. The basic time step for the time series must be truly a constant value. For example, a time series at a monthly interval does not have a constant time step. Therefore, the basic time step assigned to such a time series is daily because a day is of constant length and is comensurable with all months. The values for each month are stored in a compressed form so that each day's value need not be present on the TSS. As discussed later the daily time step is the longest basic time step available for storing information on the TSS.

1.8 Time Series Dataset Structure

Each dataset has a label describing the contents of the dataset and providing information needed for accessing the time series stored therein. Table 3 gives a description of the label structure. The length of the label is given in the TSS Descriptor.

The time series in the dataset consists of one or more members with each member having one or more components. Members are referenced by a name of up to six characters in length, e.g. "RAIN" whereas a component is referenced by a name and subscript, e.g. "RAIN 3". A reference to a member name with more than one component is taken to be a reference to all the components in the member. In the same way a reference to a time series dataset without giving member/components is taken as a reference to all member/components in the dataset. In this manner larger aggregates of time series data can be referenced concisely.

1.9 Addressing in the Dataset

The REAL words in the dataset are logically treated as if they were numbered sequentially from unity starting with the first word of the first TSS record of the dataset. The label of the dataset always begins at the first word of the dataset. The keys area of the label contains the address of the REAL word for the beginning of each calendar year stored in the dataset. The calendar years need not be stored in chronological order but the data within each calendar year is stored in chronological sequence. Access to time series information then takes place in two steps: the direct step to the TSS record containing the first word of the calendar year in question and the search to find the time interval within the year. This approach has been used to simplify the keying scheme and also to take advantage of the characteristics of use of the TSS. Normally simulation and modeling with time series involves starting at a given time and then proceeding sequentially to an ending time. Thus direct addressing with fine resolution is not needed. The search takes place within one calendar year and then only at the start of the run. Creation of the data in the dataset may require more frequent searching but should not negate the advantages of the current keying system.

1.10 Time Series Structure

The time series data in a dataset are stored in variable length blocks, called calendar year blocks (CYB), for each calendar year. The data contained within each CYB is stored in variable length blocks called time series blocks (TSB). Each TSB consists of a block control word (BCW) or a BCW and two or more time frames. A time frame contains the values for all member/components of the time series for a time point. A time frame therefore represents a section across all components of the time series. The block control word has several functions and therefore several meanings. Its functions are as follows:

1. Describe the TSB and the number of time frames in the TSB.
2. Describe the address of the next chronological year of data.
3. Indicate the end of chronological data.

The meanings for the BCW are all derived from the BCW alone and are not dependent on the position of the BCW. Each CYB in the dataset starts with a REAL word giving the year of the block but this word is not a block control word because its interpretation is dependent on its position in the dataset.

The first function of the BCW treats the word as logically representing two integers. The first integer, the Block Type Indicator (BTI), defines the block type: uncompressed, zero compressed, and undefined compressed. The second integer, the number of values (NOV), gives the number of time frames in the TSB. The last two functions of the BCW treat the word as representing a single integer. Function 2 is signalled by a negative value in the BCW and the absolute value gives the address of the next years data. A zero value for the block control word implies the third function given above.

A TSB always represents a span of time and must contain values to define the variation of both point and mean value time series over that time span. This requires an additional time point for point value time series to store the initial value for the time span of the TSB. No such point is required for mean value time series. However, space for this initial point is allocated for a TSB, unless the TSB consists of a BCW only, because point and mean value time series can be mixed in a dataset. Thus, the first frame in a TSB always contains the initial values for the point value time series in the dataset. Values in the initial frame are undefined for mean value time series. The point values stored in the initial frame of a TSB are then the same as the point values in the final frame of the predeeding TSB. The requirement for representing a span of time is motivated by the desire to have a TSB meaningful in all cases without requiring information stored in another TSB.

1.11 Addressing Conventions

The structure of the time series in the dataset requires several addresses. These are:

1. The address of the calendar year block.

2. The address of a TSB (same as the address of its BCW).

3. The address of a time frame.

4. The address of a particular component in the time frame.

The first three addresses will be given as the virtual origin of the entity in question. Each entity (CYB,TSB,frame) consists of one or more elements (words) and these words are logically numbered like the elements of a vector starting (as in Fortran) at unity. The virtual origin of a vector is the address of its zeroth element whether that element exists or not. A virtual origin as an address then leads to the convention that the offset to a particular element must always be added to the virtual origin to compute the address of the element. The last address in the list above is then given as the offset of the component from the address (virtual origin) of the time frame.

The method of storing the data into calendar year blocks, then into sequentially organized TSB's, and then into sequentially organized time frames requires that a calender year be fully represented unless it is the last calander year (in physical sequence) entered into the dataset. This last year entered, however, must be completed by the system or by the user before any other calendar year can be entered into the dataset. The system will complete the calendar year with user selected filler values.

1.12 Nature of Compression

There are two forms for the data in a TSB: uncompressed and compressed. The uncompressed form maintains space for all values in the time span of the time series. This form takes the most space and is necessary for making major changes to the data already on the TSS. Changes can always be made because there is space for all values. The compressed form suppresses repeated zero/undefined values to conserve space. The TSB's for the compressed form consist of the BCW only.

A dataset can only contain the following combinations of TSB's:

1. Uncompressed TSB's only
2. Uncompressed TSB's and compressed TSB's

Thus the nature of the compression for the dataset is determined at the dataset level. Changes in the level of compression can only be done by copying the data to another dataset. Thus, once set, the nature of the compression, specified in the dataset label, cannot be changed by a simple update.

1.13 Access Methods

There are three different access methods for writing data into a dataset:

1. ADD: This option preserves pre-existing data preceding the span of time of the current run but destroys all pre-existing data subsequent to the starting point of the current run including data after the span of time of the run. All pre-existing data in the dataset is lost if the time span of the run precedes the start of data in the dataset. The calendar year blocks in the dataset must be in physically sequential order for this option.

2. REPLACE: This option preserves pre-existing data both before and after the time span of the current run. Note that the data must be uncompressed for this access option. The number, size, and composition of the calendar year blocks, TSB's and frames will not be altered by a REPLACE option. Only data items stored in the dataset can be changed.

3. INSERT: This option writes data for calendar years which are not present in the dataset. No pre-existing data is changed or overwritten. The pre-existing data need not be in chronological order.

When you are dealing with a multi-component TSS dataset, you must heed the warnings given at the end of Section 4.6, Part F.

The least complex option is ADD and the most complex option is REPLACE. Run times and cost will reflect these differences in complexity.

TABLE 1

TSS Descriptor Contents

<-IDENTIFIER STRUCT->	TYPE	<-----------DESCRIPTION----------------------------->
1 TSSDES(8)	REC	TSS descriptor(uses 8 REAL words of space)
2 FILESZ	I4	Number of TSS records in the TSS
2 TDFREC	I4	Record number of the first TSS record in the TSS directory
2 TDLREC	I4	Record number of the last TSS record in the TSS directory
2 FREESP	I4	Number of free TSS records in the TSS
2 TDDS	I2	Dataset number of the TSS directory-default:TDDS=1
2 TOTDS	I2	Total number of datasets allowed for this TSS
2 FDS	I2	Dataset number of the first dataset in the TSS in sequence of increasing TSS record numbers. Default will be FDS=1(The TSS directory is first in the TSS by default.)
2 RECLT	I2	TSS record length in REAL words
2 FIXLBL	I2	Length of a dataset label in REAL words
2 TDSIZE	I2	Record length in real words for the TSS directory dataset
2 TSSTYP	I2	Type code for TSS for compatiability checking
2 IDUM	I2	Space holder

TABLE 2

TSS Directory Contents

<-IDENTIFIER STRUCT->	TYPE	<----------DESCRIPTION---------------------------->
1 TDREC(5)	REC	TSS directory record(length in REAL words). Space for one record for each potential dataset(including the TSS directory) is made available in the TSS directory. However, only the records for the current datasets in the TSS contain meaningful information. The records are indexed by dataset number.
2 DSNO	I2	Dataset number
2 SECURE	I2	Write protect flag. Must be OFF to write to the dataset.
2 FREC	I4	Number of the first TSS record for the dataset.
2 LREC	I4	Number of the last TSS record for the dataset.
2 FPNT	I4	Dataset number of the dataset forward of this dataset in the direction of increasing TSS record numbers.
2 BPNT	I4	Dataset number of the dataset backward of this dataset in the direction of decreaseing TSS record number.

TABLE 3

Time Series Dataset Label

<-IDENTIFIER STRUCT->	TYPE	<----------DESCRIPTION---------------------------->
1 DSLABL(185)	REC	Dataset label(length in REAL words)
2 DSLEV		All values at the dataset level
3 DSDSNO	I2	Dataset number
3 LBLSZ	I2	Label size in REAL words
3 DSFREC	I4	Record number of the first TSS record in the dataset
3 DSLREC	I4	Record number of the last TSS record in the dataset
3 VOFRWD	I4	Virtual origin of first free word in the dataset
3 VOYEAR	I4	Virtual origin of year used for updating label
3 LASTYR	I2	Last chronological year stored in the dataset

		with value of zero if no data is present
3 DSSEC	I2	Write protect flag
3 BASEYR	I4	Base year for the keying system
3 VOKEY	I4	Virtual origin for next set of VOKEY and KEYS Multiple keys feature not yet implemented
3 KEYS(100)	I4	Virtual origin of each calendar year
3 NAME(3)	I2	Name for the dataset(6 characters permitted).
3 UNITS	I2	Code giving the units of measurement of the data
3 DSCMPR (COMPR)	I2	Code for dataset compression: 1-uncompressed 2-zero and/or undefined compressed
3 YEAROR	I2	Flag to indicate that the data is stored in chronological year order.
3 OBSTIM	I2	Observation time
3 STA(3)	I2	Station name
3 LOCATN(10)	I4	Description of the location of the data source
3 GAPCOD	I2	Code for handling of any leading or trailing gaps in a calendar year for an uncompressed dataset. If ABS(GAPCOD)= 1 then both leading and trailing gaps filled with uncompressed TSB's 2 then trailing gaps filled using compressed TSB's; leading gaps filled using uncompressed TSB's 3 then leading gaps filled using compressed TSB's; trailing gaps filled using uncompressed TSB's 4 then both leading and trailing gaps filled using compressed TSB's Default is GAPCOD= 1 If GAPCOD > 0 then the time frames in the gaps are set to zero. If GAPCOD < 0 then the time frames in the gaps are set to undefined. All leading and trailing gaps are compressed if the dataset is compressed.
3 DSDELT (DELTAT)	I2	Value of the time step in minutes(limit 1440)
3 NMEMS	I4	Number of members in the dataset
3 MEMNAM(3,20)	I2	Member name(limit of 6 characters)
3 MSUB(20)	I2	Number of components
3 MKIND(20)	I2	Kind of time series(point=1 or mean=2 value)
3 FMT(20)	I2	Format code for each member

2.0 USER'S CONTROL INPUT FOR PROGRAM NEWTSS

The conventions used in this part of the documentation are the same as those used to document the User's Control Input for the HSPF program itself (Part F).

2.1 Layout

```
********************************************************************************
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
********************************************************************************

*** To create a TSS:
OPNTSS
  TSS FILE NO=     <fno>
  RECORD LENGTH=   < rl>
  MAX. DSNO=       < nd>
  TSS FILE LENGTH= <len>
  *** Optional parameter (default=1):
  DIRECTORY NO=    <dir>


*** To copy a TSS:
COPY
  FROM FILE NO=f1
  TO FILE NO=  f2

END (must appear only once in this input, at the end)
********************************************************************************
```

2.2 Details

Symbol	Fortran Name(s)	Format	Comment
<fno>	TSSFL	I5	The Fortran unit no. of the TSS. Default 23. Valid range 23 to 30.
< nd>	TOTDS	I5	The total number of datasets to be stored in the TSS directory.
< rl>	RECLT	I5	Record length of the TSS, in 4-byte words. No default, range 8 to 1000.
<len>	FILESZ	I5	The number of records in the TSS.

			(RECLT and FILESZ must agree with length spec. in DEFINE FILE statement, if your compiler requires it).
<dir>	TDDS	I5	Index number of the TSS directory. Default 1, range 1 to 32767.
f1	OLDTSS	I2	Fortran unit no. of the source TSS. Default 15. Range 15 to 22.
f2	NEWTSS	I2	Fortran unit no. of the new (target) TSS. Default 23. Range 23 to 30.

2.3 Explanation

Every input stream must end with the END keyword; only one END may be used, even if the input stream contains both an OPNTSS and a COPY operation. Each number must be right-justified in its field. In the OPNTSS option, all numbers must end in column 24; in the COPY option all numbers must end in column 17.

In the COPY option, the target TSS must be empty but already created.

Some Fortran compilers (eg. IBM 370) require that all direct access files be specified in a DEFINE FILE statement and that this statement must specify the length of the file (in records) and the record length. To assist users in this situation, the NEWTSS program has defined 8 possible files for the OLDTSS, and a corresponding set of 8 possible files for NEWTSS or TSSFL. The DEFINE FILE statements for the 8 files differ only in the number of records they specify. Details are:

Length (records)	OLDTSS Fortran unit number	NEWTSS or TSSFL Fortran unit number
48	15	23
96	16	24
240	17	25
960	18	26
4800	19	27
22500	20	28
45000	21	29
90000	22	30

Thus, to create a TSS, the user must select a Fortran unit no. for TSSFL which has the required length (or more). Similarly, when copying to a new TSS, the Fortran unit no. for the NEWTSS must agree with the length of that TSS.

2.4 Sample NEWTSS Input

Example 1. Creation of a TSS

```
OPNTSS
  TSS FILE NO=        23
  RECORD LENGTH=     512
  MAX. DSNO=          30
  DIRECTORY NO=       10
  TSS FILE LENGTH=    48
END
```

The TSS FILE LENGTH states that the length of this TSS is only 48 records. The RECORD LENGTH field states that each record contains 512 words (2048 bytes), which is a suitable configuration for both HP3000 and IBM370 systems. Note that users on those installations should not use any other value for the record length, unless the DEFINE FILE, and many other, statements in the code have been suitably altered.

The user has requested that the directory have space for 30 datsets. The NEWTSS program may actually configure the TSS with directory space for more than this number, if this does not require addtional space for the directory. When this happens an informative message is printed.

The DIRECTORY NO field states that the directory is to occupy dataset no 10.

The TSS FILE LENGTH field must agree with the length implied by the TSS FILE NO field, if the compiler imposes the restrictions discussed earlier.

Example 2. Copying of a TSS

```
COPY
  FROM FILE NO=15
  TO FILE NO=  23
END
```

2.5 Estimating the Size Required for a TSS

The space required by a TSS, expressed in terms of records of 512-word (2048 byte) length is:

Space = 1+ CEIL(MAXDSNO/512) + (MAXDSNO-1) + SIGMAi(Pi*Ci*Yi*Fi)

where:

MAXDSNO	is the maximum dataset number of any dataset that will be stored in the TSS. This number determines the size of the TSS directory.
CEIL(x)	is the smallest integer greater than or equal to x.
SIGMAi	means the sum for all values of i.
Pi	is the proportion of full uncompressed space used by the time series data in the i-th dataset. It is 1 for uncompressed datasets and less than 1 for compressed datasets.
Ci	is the number of components to be stored in the i-th dataset.
Yi	is the number of calendar years of data to be stored in the i-th dataset (part of a calendar year counts as a complete year, with the exception discussed below).
Fi	is the time step factor, obtained from the table below.

Note that, even for an uncompressed dataset, it is possible to compress part of the first calendar year, if data will never be stored in that part (eg. Jan thru June). This is called a "leading gap". It is also possible to compress a "trailing gap" (Section 2.3, Part F). If this is done, the compressed parts of those years can be eliminated from the space computed using the above formula.

The space indicated by this formula should be increased by at least 10% to allow for unforseen requirements. Note, however, that it is possible to copy the contents of a TSS to a larger TSS, if you run out of space.

Esimating the Space Required by Each Dataset in the TSS

The space required for the i-th dataset, in records, is:

Space=   1 + Pi*Ci*Yi*Fi

where the terms are as previously defined (the adjustment for compressed leading and/or trailing gaps - discussed above - also applies here).

Space Requirements of Uncompressed Data, in 512-word Records

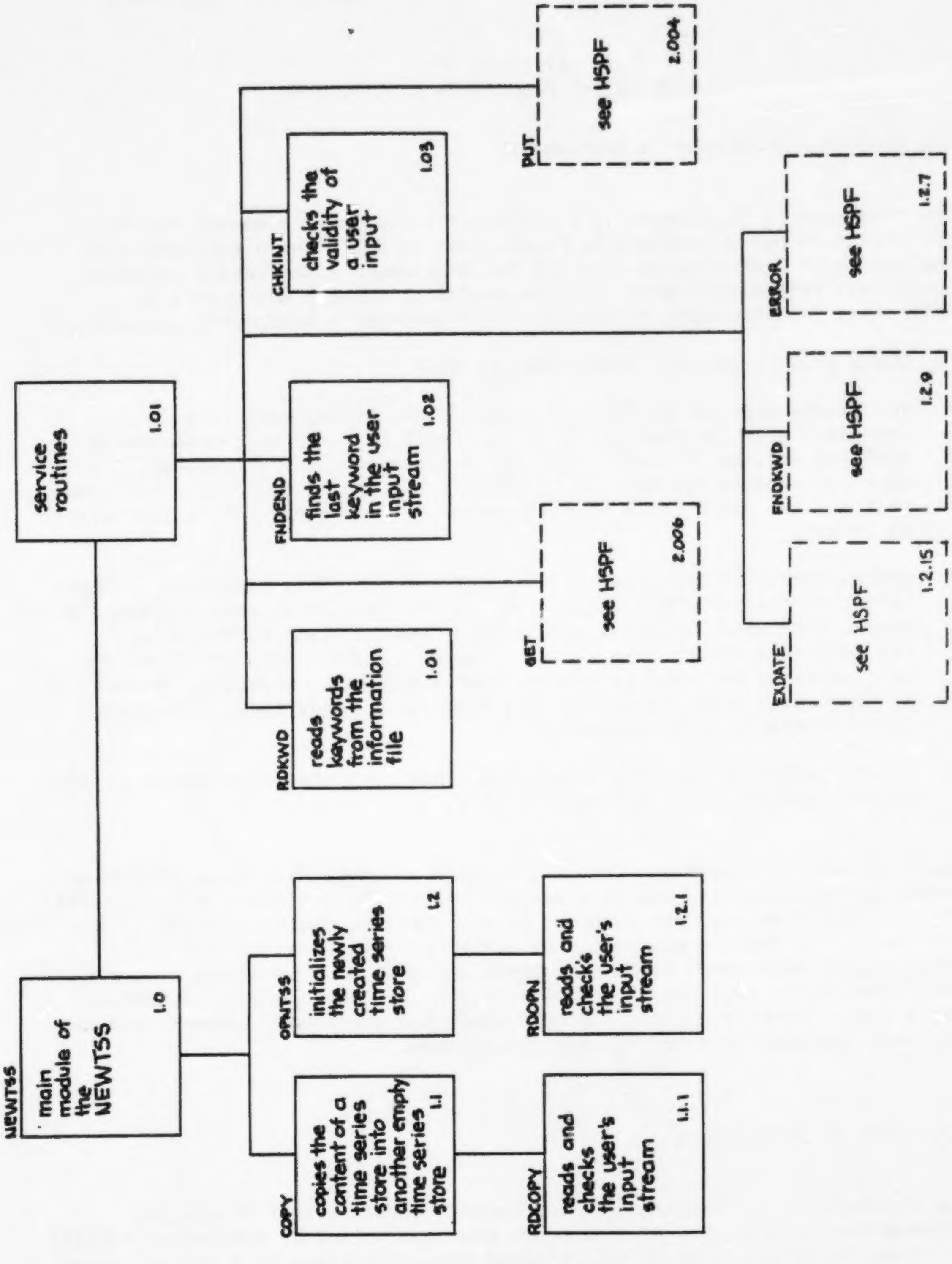
Time-step (mins)	Time-step factor Records/yr	Time-step (mins)	Time-step factor Records/yr
1	1040	20	52.0
2	520	30	34.7
3	347	60	17.4
4	260	120	8.7
5	208	180	5.8
6	174	240	4.4
10	104	360	3.0
12	86.8	480	2.2
15	69.4	720	1.5
		1440	0.75

Example. Given:

1. Seven datasets at hourly interval, 7 years each
2. Four datasets at daily interval, 7 years each
3. Max. dataset number= 20
4. All datasets uncompressed and with a single component

```
TSS Space=  1 + CEIL(20/512) + (20-1) + 7(1*1*7*17.4) + 4(1*1*7*.75)
         =  1 + 1            + 19     + 852           + 21
         =  894
```

Increase by 10% to, say, 1000 records.


NEWTSS
main module of the NEWTSS
1.0
service routines
1.01
COPY
copies the content of a time series store into another empty time series store
1.1
OPNTSS
initializes the newly created time series store
1.2
RDCOPY
reads and checks the user's input stream
1.1.1
RDOPN
reads and checks the user's input stream
1.2.1
RDKWD
reads keywords from the information file
1.01
FNDEND
finds the last keyword in the user input stream
1.02
CHKINT
checks the validity of a user input
1.03
GET
see HSPF
2.006
PUT
see HSPF
2.004
EXDATE
see HSPF
1.2.15
FNDKWD
see HSPF
1.2.9
ERROR
see HSPF
1.2.7

Structure chart 1.0 The NEWTSS program

APPENDIX IV
GUIDE TO THE PROGRAMMER'S SUPPLEMENT

1.0 About the Programmer's Supplement

The Programmer's Supplement is a companion to the user's manual for HSPF. The user's manual describes the foundations of HSPF, the algorithms used, and the input needed to operate it; the Programmer's Supplement contains additional information which will be useful to someone who wishes to understand its structure in detail. The Programmer's Supplement consists of:

1. Lists showing all the subprograms in HSPF.

2. The source code in the form of pseudo code. Pseudo code is a non-compilable language which lends itself to structured programming. HSPF was written in pseudo code and translated into fortran. While a user may examine the fortran by dumping the source code tape to a line printer, the pseudo code is much easier to understand, being "narrative" in format.

3. Documentation of the principal data structures. This part describes the layout of the variables and arrays and the data which they represent or store. HSPF makes use of a single large common block to store data in the machine's memory. The various modules in HSPF each have their own configuration for data in common. When a module is inactive, its version of the common block is stored on a disc file, ready to be copied back when the module is re-activated.

4. Documentation of the file structures. This part shows the layout of data on the various disc files used by HSPF.

The Programmer's Supplement is not a printed document, but consists of text stored in datasets recorded on a magnetic tape. The intention is to include these datasets on the tape supplied to any user who requests a copy of the HSPF software. Then he can easily obtain a reference copy of the Programmer's Supplement by dumping these datasets to a hard-copy device such as a terminal or line printer. To obtain a highly legible copy, he should use a device capable of printing both upper and lower case letters, because the text has been recorded in upper/lower case.

2.0 LIST OF SUBPROGRAMS

Two datasets in the Programmer's Supplement contain a list of all the subprograms in HSPF. The first dataset has them sorted by subprogram number; it is useful if you need to scan through the subprograms in a logical order.

The numbering system in HSPF is based on the hierarchical arrangement of the subprograms. The system is discussed in Part C of this document, and illustrated in the Visual Table of Contents (Part D). The second dataset has them listed in alphabetical order (by subprogram name); it is useful if you need to locate, or find the number for, a subprogram when you know its name.

These datasets have been included in the Programmer's Supplement because there are over 500 subprograms in HSPF; locating one of them without the help of an index could be quite a problem.

3.0 PSEUDO CODE

Pseudo code is a non-compilable language for writing computing algorithms. It is much better suited to the writing of well-structured algorithms than a language such as Fortran, because it uses the five basic "structure figures" needed to write structured programs:

1. Sequence of elementary statements
2. IFTHENELSE
3. WHILEDO
4. DOUNTIL
5. CASE

These structure figures are discussed in Part C of this document, as are other details of the pseudo code language and the methods used to translate it into Fortran.

HSPF was written in pseudo code and then translated into Fortran. Because pseudo code is much easier to read than Fortran, once one is familiar with it, we recommend that a person interested in studying the HSPF algorithms read the pseudo code rather than the Fortran. Note however, that anyone who wishes to read the Fortran code may do so by dumping the Fortran source code datasets to a hard-copy device such as a line printer, or by reading listings produced by his Fortran compiler.

Following the tenets of Structured Program Design (Part C) the subprograms in HSPF are arranged in hierarchical order and are numbered accordingly. The arrangement is depicted in the Visual Table of Contents (Part D). The pseudo code follows the same organization; subprograms are arranged in numerical order, which means that they follow a logical, rather than an alphabetical, pattern. We start at the highest level of the system (MAIN program) and proceed down the various branches of the tree to the more detailed levels, covering the system in a top-down, left-to-right sequence.

4.0 STRUCTURE OF DATA IN CORE

HSPF makes use of a single, large COMMON block (called SCRTCH) to store data in the machine's memory. The various modules in HSPF each have their own version of this block, so that the arrangement of data in SCRTCH can be tailored to their individual needs. For example, the application modules PERLND, IMPLND and RCHRES all use the first part of SCRTCH to store their respective parameters, state variables and computed fluxes, while the latter part is used to hold time series (input and computed). On the other hand, the Run Interpreter does not deal with time series, so uses the latter part of SCRTCH to store the various tables which it uses to process the User's Control Input.

There are, thus, many different versions of COMMON block SCRTCH in HSPF, each with its own arrangement of variables and arrays.

HSPF has been designed so that, when a module such as PERLND is temporarily interrupted to permit some other module to perform its work, the data in SCRTCH is copied to disc so that it can be copied back when PERLND resumes execution. Meanwhile, the other module (say RCHRES) user SCRTCH to store its data, and also copies the contents to disc when it is interrupted and PERLND resumes operation.

In the HSPF data structures we have made liberal use of the EQUIVALENCE statement to overlay data. For example, a pair of state variables in the PERLND module, INFFAC and PETADJ, are equivalenced to an array PWST3(2). In the HSPF code, the former names are used when the individual variables need to be referenced and the latter "group name" is used when both variables are being dealt with together.

The data structures used in HSPF are documented in the Programmer's Supplement in a set of datasets with names commencing with DATSTR. These datasets describe:

1. The arrangement of variables and arrays, including group names (implemented in Fortran using the EQUIVALENCE statement).
2. The type (eg. R4,I2) of each variable.
3. The function of each variable (a short description of what it represents).
4. The "kind" of each variable which represents a time series (point-valued or mean-valued)
5. The internal and external units, in the English and Metric systems, where applicable.

5.0 STRUCTURE OF DATA FILES

The HSPF system makes use of several different classes of disc-based files:

1. The Time Series Store (TSS), which is documented in Section 2, Part E and in Appendix III.

2. Files containing the User's Control Input: USRFL is a sequential card-image file on which the user submits his input. UCIFL is a direct access file onto which HSPF transfers the User's Control Input. Each record contains data from the corresponding record in USRFL, plus "chaining" information which points to the next non-comment line in the input. This is the file on which HSPF operates when it sorts and analyses the User's Control Input.

3. A "plot-file". The PLTGEN module uses this file to store its output - up to 10 time series which are to be simultaneously displayed by a plotter. This file is documented in Section 4.2(12), Part E.

4. The information file (INFOFL), error message file (ERRFL) and warning message file (WARNFL). These files are self-documenting. One need only list the file and read it to understand its contents. Users will probably not need to read the INFOFL. It supplies information to the HSPF code which enables it to process the User's Control Input; the same information is contained in Part F of this manual, in more readable form.

5. Instruction files. The Operations Supervisor Instruction File (OSUPFL), TSGET Instruction File (TSGETF) and TSPUT Instruction File (TSPUTF) contain instructions, generated by the Run Interpreter, which govern the execution of HSPF. They are documented in the Programmer's Supplement.

6. Operation Status Vector File (OSVFL). This file stores the contents of COMMON block SCRTCH so that a module can resume execution with the same values for parameters and state variables as it had when execution was interrupted (see Section 4.0 above). It is documented in the Programmer's Supplement.

7. Working file (WORKFL). The Run Interpreter uses this as a scratch file, to store potentially large tables of information generated while analyzing and sorting the User's Control Input. It is documented in Programmer's Supplement.